Matemática Discreta e Suas Aplicações

SEXTA EDIÇÃO

Kenneth H. Rosen

Mc Graw Hill

VISIT…

Matemática Discreta e Suas Aplicações

Sexta Edição

Kenneth H. Rosen
AT&T Laboratories

Tradução Técnica

Helena Castro
Professora Doutora do Instituto de Matemática e Estatística da Universidade de São Paulo – IMEUSP.

Professor João Guilherme Giudice
Matemático pela Universidade Estadual de Campinas – UNICAMP. Doutorando em Filosofia na área de lógica matemática pela Universidade Estadual de Campinas – UNICAMP. Atua como professor em instituições particulares de ensino.

Versão impressa
desta obra: 2009

AMGH Editora Ltda.
2010

The McGraw-Hill Companies

Matemática Discreta e Suas Aplicações
Sexta edição
ISBN: 978-85-7726-036-2

A reprodução total ou parcial deste volume por quaisquer formas ou meios, sem o consentimento, por escrito, da editora, é ilegal e configura apropriação indevida dos direitos intelectuais e patrimoniais dos autores.

©2009 McGraw-Hill Interamericana do Brasil Ltda.
Todos os direitos reservados.
Av. Brigadeiro Faria Lima, 201, 17º andar.
São Paulo – SP – CEP 05426-100

©2009 McGraw-Hill Interamericana Editores, S.A. de C.V.
Todos os direitos reservados.
Prol. Paseo de la Reforma 1015 Torre A
Piso 17, Col. Desarrollo Santa Fé,
Delegación Alvaro Obregón
C.P. 01376, México, D.F

Tradução da sexta edição em inglês de *Discrete Mathematics and Its Applications*.
Publicada pela McGraw-Hill, uma unidade de negócios da McGraw-Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY 10020.
©2007 by Kenneth H. Rosen
ISBN da obra original: 978-0-07-288008-3

Coordenadora Editorial: *Guacira Simonelli*
Editora de Desenvolvimento: *Marileide Gomes*
Produção Editorial: *RevisArt*
Supervisora de Pré-impressão: *Natália Toshiyuki*
Preparação de Texto: *Patrícia Travanca/Marise Goulart*
Design de Capa: *Rebidas Design & Associates*
Imagem de Capa: Cover art ©Jasper Johns (b. 1930)/Autorizado por VAGA, New York, NY. *Dancers on a Plane*.
Diagramação: Crontec Ltda.

R813m Rosen, Kenneth H.
Matemática discreta e suas aplicações [recurso eletrônico] / Kenneth H. Rosen ; tradução técnica: Helena Castro, João Guilherme Giudice. – 6. ed. – Dados eletrônicos. – Porto Alegre : AMGH, 2010.

Editado também como livro impresso em 2009.
ISBN 978-85-63308-39-9

1. Matemática. I. Título.

CDU-501

Catalogação na publicação: Ana Paula M. Magnus – CRB-10/Prov-009/10

Sumário

Sobre o Autor vi
Prefácio vii
Ao Estudante xix

1 Os Fundamentos: Lógica e Demonstrações 1

1.1 Lógica Proposicional 1
1.2 Equivalências Proposicionais 21
1.3 Predicados e Quantificadores 30
1.4 Quantificadores Agrupados 50
1.5 Regras de Inferência 63
1.6 Introdução a Demonstrações 75
1.7 Métodos de Demonstração e Estratégia 86
Termos-chave e Resultados 104

2 Estruturas Básicas: Conjuntos, Funções, Seqüências e Somatórios 111

2.1 Conjuntos 111
2.2 Operações com Conjuntos 121
2.3 Funções 133
2.4 Seqüências e Somatórios 149
Termos-chave e Resultados 163

3 Os Fundamentos: Algoritmos, Números Inteiros e Matrizes ... 167

3.1 Algoritmos 167
3.2 O Crescimento de Funções 180
3.3 Complexidade dos Algoritmos 193
3.4 Os Números Inteiros e a Divisão 200
3.5 Os Números Primos e os Máximos Divisores Comuns 210
3.6 Os Números Inteiros e os Algoritmos 219
3.7 Aplicações da Teoria dos Números 231
3.8 Matrizes 246
Termos-chave e Resultados 257

4 Indução e Recursão 263

4.1 Indução Matemática 263
4.2 Indução Completa e Boa Ordenação 283
4.3 Definições Recursivas e Indução Estrutural 294
4.4 Algoritmos Recursivos 311
4.5 Exatidão de Programas 322
Termos-chave e Resultados 328

5 Contagem 335

5.1 As Bases da Contagem 335
5.2 O Princípio da Casa dos Pombos 347
5.3 Permutações e Combinações 355
5.4 Coeficientes Binomiais 363
5.5 Permutações e Combinações Generalizadas 370
5.6 Gerando Permutações e Combinações 382
Termos-chave e Resultados 386

6 Probabilidade Discreta 393

6.1 Uma Introdução à Probabilidade Discreta 393
6.2 Teoria da Probabilidade 400
6.3 Teorema de Bayes 417
6.4 Valor Esperado e Variância 426
Termos-chave e Resultados 442

7 Técnicas Avançadas de Contagem 449

7.1 Relações de Recorrência 449
7.2 Resolvendo Relações de Recorrência Lineares 460
7.3 Algoritmos de Divisão e Resolução e Relações de Recorrência 474
7.4 Funções Generalizadas 484
7.5 Inclusão–Exclusão 499
7.6 Aplicações da Inclusão–Exclusão 505
Termos-chave e Resultados 513

8 Relações 519

8.1 Relações e Suas Propriedades 519
8.2 Relações *n*-árias e Suas Aplicações 530
8.3 Representação de Relações 537
8.4 Fecho de Relações 544
8.5 Relações de Equivalência 555
8.6 Ordens Parciais 566
Termos-chave e Resultados 581

9 Grafos 589

9.1 Grafos e Modelos de Grafos 589
9.2 Terminologia dos Grafos e Tipos Especiais de Grafos 597
9.3 Representação de Grafos e Isomorfismo de Grafos 611
9.4 Conectividade 621
9.5 Caminhos Eulerianos e Hamiltonianos 633
9.6 Problemas de Caminho mais Curto 647
9.7 Grafos Planares 657
9.8 Coloração de Grafos 666
Termos-chave e Resultados 675

10 Árvores 683

10.1 Introdução às Árvores 683
10.2 Aplicações de Árvores 695
10.3 Percurso em Árvores 710
10.4 Árvores Geradoras 724
10.5 Árvores Geradoras Mínimas 737
Termos-chave e Resultados 743

11 Álgebra Booleana 749

11.1 Funções Booleanas 749
11.2 Representação de Funções Booleanas 757
11.3 Portas Lógicas 760
11.4 Minimização de Circuitos 766
Termos-chave e Resultados 781

12 Modelagem Computacional 785

12.1 Linguagens e Gramáticas 785
12.2 Máquinas de Estados Finitos com Saída 796
12.3 Máquinas de Estados Finitos sem Saída 804
12.4 Reconhecimento de Linguagem 817
12.5 Máquinas de Turing 827
Termos-chave e Resultados 838

Apêndices 845

A–1 Axiomas para os Números Reais e para os Inteiros Positivos 845
A–2 Funções Exponenciais e Logarítmicas 851
A–3 Pseudocódigo 854

Sugestões de Leitura **861**

Respostas dos Exercícios de Número Ímpar **869**

Créditos das Fotos **959**

Índice de Nomes **961**

Índice Remissivo **963**

Sobre o Autor

Kenneth H. Rosen tem uma longa carreira como Membro Renomado da Equipe Técnica dos Laboratórios AT&T em Monmouth County, Nova Jersey. Ele mantém atualmente a posição de professor pesquisador convidado na Universidade de Monmouth, onde trabalha com aspectos de segurança e privacidade no Projeto de Resposta Rápida em Database e também leciona um curso sobre aplicações criptográficas.

Dr. Rosen recebeu seu diploma de bacharel em matemática na Universidade de Michigan, em Ann Arbor (1972), e seu Ph.D. em matemática no M.I.T. (1976), onde escreveu sua tese na área da teoria dos números sob a orientação de Harold Stark. Antes de ingressar nos Laboratórios Bell, em 1982, ele ocupou cargos na Universidade do Colorado, em Boulder; na Universidade do Estado de Ohio, em Columbus; e na Universidade de Maine, em Orono, onde foi professor associado de matemática. Enquanto trabalhava nos Laboratórios AT&T, ele ministrou, na Universidade de Monmouth, cursos de matemática discreta, teoria de código e segurança de dados.

Dr. Rosen publicou inúmeros artigos em revistas acadêmicas nas áreas de teoria dos números e modelos matemáticos. Ele é autor dos livros *Teoria dos Números Elementares e Suas Aplicações*, publicado pela editora Addison-Wesley e atualmente em sua quinta edição, e *Matemática Discreta e Suas Aplicações*, publicado pela McGraw-Hill, atualmente na sexta edição. Ambos os livros são muito usados por centenas de universidades em todo o mundo. *Matemática Discreta e Suas Aplicações* teve mais de 300 mil exemplares vendidos, com tradução para espanhol, francês, chinês e coreano. Ele também é co-autor de *UNIX: A Referência Completa; Sistema UNIX versus Release 4: Uma Introdução;* e *Os melhores tipos da UNIX*, todos publicados pela editora de Osborne McGraw-Hill, cujas vendas foram superiores a 150 mil cópias, com tradução para chinês, alemão, espanhol e italiano. Dr. Rosen também é editor do *Livro de Bolso de Matemática Discreta e Combinatória,* publicado pela CRC Press, e é o editor consultivo da série CRC de livros de matemática discreta, que contém mais de 30 volumes sobre diferentes aspectos em matemática discreta, dos quais a maioria está inserido neste livro. Ele também se interessa por software de integração matemática no âmbito educacional e profissional e trabalhou nessas áreas em projetos com o software Maple, da Waterloo Maple Inc.

Nos Laboratórios Bell e AT&T, Dr. Rosen trabalhou em uma grande variedade de projetos, incluindo pesquisas operacionais e planejamento de linha de produção para computadores e equipamentos de comunicação. Ajudou no planejamento de produtos e serviços da AT&T na área de multimídia, incluindo comunicação em vídeo, reconhecimento e síntese da fala e transmissão de imagem. Ele avaliou a nova tecnologia para uso da AT&T e iniciou os trabalhos na área de transmissão de imagens. Também desenvolveu diversos novos serviços, mantendo ou submetendo mais de 70 patentes. Um de seus projetos mais interessantes envolveu a ajuda de avaliação tecnológica para a atração da AT&T no EPCOT Center.

Prefácio

Ao escrever este livro, fui guiado pela minha longa experiência e pelo interesse em ensinar matemática discreta. Para o estudante, minha proposta foi oferecer um material que fosse preciso e de fácil leitura, com conceitos e técnicas da matemática discreta claramente apresentados e demonstrados. Meu objetivo foi mostrar sua relevância e praticidade aos estudantes, que geralmente são céticos. Quis dar-lhes um estudo sobre a ciência da computação com todos os fundamentos da matemática de que eles precisarão em estudos futuros. Quis passar uma compreensão da importância dos conceitos matemáticos com uma idéia do porquê esses conceitos são importantes para as aplicações. E o mais importante, quis alcançar esses objetivos sem diluir o conteúdo.

Para o professor, minha proposta foi criar uma ferramenta de ensino flexível e compreensível, que usa técnicas pedagógicas aprovadas em matemática. Quis dar-lhes um material que eles pudessem usar para ensinar efetivamente matemática discreta, de maneira eficiente e a mais apropriada possível, para os mais diferentes níveis de estudantes. Espero ter alcançado esses objetivos.

Estou imensamente grato pelo grande sucesso deste livro. As muitas melhorias nesta edição[1] tornaram-se possíveis pelas revisões e pelas sugestões de muitos professores e estudantes de mais de 600 escolas norte-americanas onde este livro tem sido utilizado com muito sucesso. Há muito empenho nesta edição.

Este livro foi elaborado para um ou dois semestres do curso introdutório à matemática discreta para estudantes das mais variadas áreas, incluindo matemática, ciência da computação e engenharia. O único pré-requisito explícito é álgebra de nível universitário, embora seja necessário determinado grau de maturidade matemática para estudar a matemática discreta de maneira significativa.

Objetivos de um Curso de Matemática Discreta

Um curso de matemática discreta possui mais de uma proposta. Os estudantes devem aprender um conjunto particular de fatos matemáticos e saber aplicá-los; o mais importante é como um curso deve ensinar os alunos a pensar lógica e matematicamente. Para alcançar esses objetivos, este livro permeia a argumentação matemática e as diversas formas de resolver os problemas. Cinco temas importantes são abordados: argumentação matemática, análise combinatória, estruturas discretas, pensamento algorítmico e aplicações e modelos. Um curso de matemática discreta completo deve unir e balancear cuidadosamente esses cinco temas.

1. *Argumentação Matemática*: Os estudantes devem entender a argumentação matemática para ler, compreender e construir argumentos matemáticos. Este livro inicia-se com uma discussão sobre a lógica matemática, que é a base para as discussões subseqüentes sobre métodos de demonstração. Tanto as ciências como a arte de desenvolver demonstrações são direcionadas. A técnica da indução matemática é desenvolvida a partir de diferentes tipos de exemplos, desde demonstrações até uma explicação cuidadosa do porquê de a indução matemática ser uma técnica de demonstração válida.
2. *Análise Combinatória*: Uma importante maneira de solucionar problemas é a habilidade de contar ou enumerar objetos. A discussão sobre enumeração começa, neste livro, com as técnicas básicas de contagem. A ênfase é dada ao desempenho da análise combinatória, a fim de resolver problemas de contagem e análise de algoritmos, e não à aplicação de fórmulas.

1 N.E.: Sexta edição norte-americana e primeira edição brasileira.

3. *Estruturas Discretas:* Um curso de matemática discreta deve ensinar os alunos a trabalhar com estruturas discretas, que são estruturas matemáticas abstratas usadas para representar objetos discretos e as relações entre eles. Essas estruturas discretas incluem conjuntos, permutações, relações, grafos, árvores e mecanismos de estado finito.
4. *Pensamento Algorítmico*: Determinadas classes de problemas são solucionadas a partir da especificação de um algoritmo. Depois que um algoritmo foi descrito, um programa de computador pode ser construído, implementando-o. As porções matemáticas desta atividade incluem a especificação do algoritmo, a verificação de seu funcionamento e a análise da memória do computador e o tempo para seu desempenho, todas abordadas neste livro. Os algoritmos são descritos usando o português e um pseudocódigo de fácil compreensão.
5. *Aplicações e Modelos*: A matemática discreta possui aplicações concebíveis para quase todas as áreas de estudo. Há muitas aplicações nesta obra para a ciência da computação e para a transmissão de dados, assim como aplicações para diversas áreas, como química, botânica, zoologia, lingüística, geografia, negócios e internet. Essas aplicações são utilizações simples e importantes da matemática discreta, e não são invenções. Modelar com matemática discreta é uma ferramenta extremamente importante na solução de problemas, a qual os estudantes têm a oportunidade de desenvolver através da construção de seus próprios modelos em alguns dos exercícios.

Alterações na Sexta Edição

A edição anterior deste livro foi utilizada por mais de 600 escolas nos Estados Unidos, dezenas de universidades canadenses e em universidades em toda a Europa, a Ásia e a Oceania. Embora ela tenha sido extremamente eficiente, muitos docentes, incluindo usuários de longa data, pediram algumas modificações para tornar o texto mais eficaz. Dediquei horas e energia significativas para satisfazer esses pedidos e trabalhei arduamente para encontrar maneiras de melhorar este livro.

O resultado é a sexta edição, que oferece muito mais do que fez a quinta edição, tanto aos docentes quanto aos estudantes. O mais significativo foi a melhora da organização dos tópicos nesta edição, que torna o livro uma ferramenta de ensino mais eficiente. Foram implementadas mudanças para melhor satisfazer tanto aos estudantes, que necessitam de toda ajuda possível, quanto àqueles que desejam ser desafiados ao máximo. Melhorias substanciais no material direcionado à lógica, ao método e às estratégias de demonstração foram projetadas para ajudar os estudantes na argumentação matemática. Foram incluídas explicações adicionais e exemplos para tornar mais claros alguns pontos em que os estudantes apresentam maior dificuldade. Foram inseridos também novos exercícios, de rotina e desafios. Aplicações de alta relevância foram adicionadas, incluindo muitas relacionadas à internet e à ciência da computação.

Organização com Melhorias

- A primeira parte do livro foi reestruturada para apresentar o conteúdo dos tópicos de maneira mais eficiente, eficaz e flexível.
- O conteúdo da argumentação matemática e da demonstração está concentrado no Capítulo 1, flutuando entre lógica proposicional e de predicados e regras de inferência, técnicas básicas e mais avançadas de demonstração e suas estratégias.
- Um capítulo separado para estruturas discretas – o Capítulo 2 desta nova edição – abrange conjuntos, funções, seqüências e somatórios.
- O material básico sobre teoria dos números, abordado em uma seção na quinta edição, é agora tratado em duas seções: a primeira com divisibilidade e congruências, e a segunda com números primos.
- O novo Capítulo 4 é inteiramente voltado a Indução e Recursão.

Lógica

- O conteúdo sobre lógica foi ampliado com as idéias-chave explicadas com grande profundidade e com maior cuidado.
- Proposições condicionais e a lei de De Morgan tiveram seu conteúdo expandido.
- A construção de tabelas-verdade foi introduzida antes e mais detalhadamente.
- Atenção especial foi dada à introdução de predicados e quantificadores, assim como à explicação de como usá-los e trabalhar com eles.
- Foi expandida a aplicação da lógica a sistemas de especificação — um tópico direcionado a sistema e engenharia de hardware e software.
- O material com argumentos e regras válidas de inferência está agora em uma seção separada.

Escrita e Entendimento de Demonstrações

- Métodos e estratégias de demonstração são tratados em seções separadas do Capítulo 1.
- Um apêndice foi adicionado com uma listagem dos axiomas básicos para os números reais e para os números inteiros e de como esses axiomas são usados para demonstrar novos resultados. O uso desses axiomas e os resultados básicos que os acompanham foram utilizados em diversas demonstrações no livro.
- O processo de criação de conjecturas, que usa diferentes métodos e estratégias de demonstração para desenvolvê-las, é ilustrado por meio de um tópico acessível sobre peças do tabuleiro de xadrez.
- O Capítulo 4 inicia-se com seções separadas e ampliadas sobre indução matemática e indução completa. Essas seções incluem mais motivação e uma rica coletânea de exemplos, muitos deles diferentes daqueles vistos normalmente.
- Mais demonstrações foram disponibilizadas, de modo que tornam possível a explicitação de todos os seus passos.

Algoritmos e Aplicações

- Uma maior abordagem foi reservada ao uso da indução completa a fim de demonstrar que algoritmos recursivos estão corretos.
- Está presente nesta edição a descrição de como o Teorema de Bayes pode ser usado para construir filtros de *spam*.
- Foram acrescentados exemplos e exercícios de geometria computacional, incluindo triangulações de polígonos.
- Foi introduzida a aplicação de grafos bipartidos para resolução de problemas.

Teoria dos Números, Análise Combinatória e Teoria da Probabilidade

- O conteúdo sobre teoria dos números está agora mais flexível, com quatro seções que abordam aspectos diferentes do assunto, sendo o conteúdo das três últimas seções opcional.
- A introdução das técnicas básicas de contagem, permutação e combinação foram melhoradas.
- O conteúdo sobre técnicas de contagem foi expandido. Agora, abrange a contagem das maneiras como podemos distribuir objetos em caixas.
- O conteúdo da teoria da probabilidade foi expandido, com a introdução de uma nova seção do Teorema de Bayes.

Grafos e Teoria da Computação

- A introdução à teoria dos grafos foi melhorada e racionalizada.
- Foi inserida uma rápida introdução à terminologia e às aplicações, com ênfase na escolha das melhores opções ao construir um modelo com grafos em vez da terminologia.
- O material sobre grafos bipartidos e suas aplicações foi ampliado.
- Foram adicionados exemplos para ilustrar a construção de máquinas automatizadas de estado finito que reconhecem conjuntos específicos.
- A minimização de máquinas de estado finito é agora mencionada e desenvolvida em uma série de exercícios.
- O conteúdo sobre máquinas de Turing foi expandido com uma breve introdução sobre como elas surgem em um estudo de complexidade computacional, decidibilidade e computabilidade.

Exercícios e Exemplos

- Foram inseridos muitos exercícios de rotina novos e exemplos, especialmente em lugares nos quais foram introduzidos conceitos-chave.
- Esforços extras foram feitos para assegurar que exercícios numerados e os extras sejam providos de conceitos básicos e habilidades.
- Melhor correspondência foi feita entre os exemplos que introduzem conceitos-chave e os exercícios de rotina.
- Muitos novos exercícios de desafio foram acrescentados.
- Mais de 400 exercícios foram inseridos com mais conceitos-chave, assim como introduzidos novos tópicos.

Biografias Adicionais, Notas Históricas e Novas Descobertas

- Biografias de Arquimedes, Hopper, Stirling e Bayes foram inseridas.
- Muitas biografias existentes na edição anterior foram melhoradas, incluindo a biografia de Augusta Ada.
- As notas históricas inseridas ao longo do livro e em notas de rodapé foram melhoradas.
- As novas descobertas feitas desde a publicação da edição anterior estão sendo apresentadas nesta edição.

Características Especiais

ACESSIBILIDADE Esta obra foi criada para ser facilmente lida e entendida por iniciantes. Não há pré-requisitos matemáticos, além da álgebra de nível universitário, para quase toda ela. Os estudantes que necessitarem de ajuda extra encontrarão ferramentas no site da MathZone (www.mathzone.com), que lhes fornecerá a maturidade matemática necessária. As informações do site estão disponíves em inglês. Os poucos espaços neste livro nos quais noções de cálculo são expressas estão anotados explicitamente. A maioria dos estudantes entenderá facilmente o pseudocódigo usado no texto para expressar algoritmos, independentemente do que é estudado formalmente na linguagem de programação. Não há pré-requisito de ciência da computação formal.

Cada capítulo inicia-se em um nível de fácil entendimento e acessível. Uma vez adquiridos cuidadosamente os conceitos matemáticos básicos, vão sendo apresentados materiais e aplicações em outras áreas de estudo mais difíceis.

FLEXIBILIDADE Este texto foi cuidadosamente criado para o uso flexível. A dependência dos capítulos anteriores foi minimizada. Cada capítulo está dividido em seções do mesmo tamanho, aproximadamente, e estas divididas em subseções que formam blocos de material para estudo. Os docentes podem facilmente dividir as leituras a partir desses blocos.

ESTILO DA ESCRITA O estilo da escrita neste livro é direto e pragmático. A linguagem matemática precisa é usada sem formalismo e abstração excessivos. É necessário tomar cuidado para balancear a mistura de notações e palavras nas proposições matemáticas.

RIGOR E PRECISÃO MATEMÁTICOS Nesta obra, todas as definições e os teoremas são propostos com muito cuidado para que os estudantes apreciem a precisão de linguagem e o rigor necessários em matemática. As demonstrações são motivadas e desenvolvidas lentamente; seus passos são todos cuidadosamente explicados. Os axiomas usados nas demonstrações e as propriedades básicas que os seguem são claramente explicados em um apêndice, dando aos estudantes uma idéia clara do que eles podem assumir em uma demonstração. As definições recursivas são também explicadas e muito utilizadas.

EXEMPLOS DESENVOLVIDOS Mais de 750 exemplos são utilizados para ilustrar conceitos, relativos a diferentes tópicos, e introduzir aplicações. Na maioria deles, uma questão é primeiramente proposta, então sua solução é apresentada com a quantidade apropriada de detalhamento.

APLICAÇÕES As aplicações apresentadas neste texto demonstram a utilidade da matemática discreta na solução de problemas reais. Este texto inclui aplicações em muitas áreas, como ciência da computação, transmissão de dados, psicologia, química, engenharia, lingüística, biologia, negócios e internet.

ALGORITMOS Os resultados em matemática discreta geralmente são expressos em termos de algoritmos; conseqüentemente, algoritmos importantes são introduzidos em cada capítulo do livro. Esses algoritmos são expressos em palavras e em estruturas de pseudocódigos facilmente compreendidos, que estão descritos e especificados no Apêndice A–3. A complexidade computacional dos algoritmos é analisada também em um nível elementar.

INFORMAÇÕES HISTÓRICAS O que está por trás de muitos tópicos é sucintamente descrito nesta obra. Foram incluídas pequenas biografias de mais de 65 matemáticos e cientistas da computação, acompanhadas de fotos ou imagens. Essas biografias incluem informações sobre as vidas, as carreiras e as realizações desses importantes pensadores que contribuíram com a matemática discreta. Além disso, há várias notas históricas, no fim da página, que foram incluídas para acrescentar informações ao contexto principal da obra. Foram feitos esforços para manter o livro atualizado, por meio da reflexão sobre as descobertas recentes.

TERMOS-CHAVE E RESULTADOS Uma lista de termos-chave e resultados acompanham todos os capítulos. Os termos-chave incluem apenas aqueles mais importantes que os estudantes devem aprender, e não todo termo definido no capítulo.

EXERCÍCIOS Há mais de 3,8 mil exercícios neste livro, com diversos tipos de questões. Há também ampla provisão de exercícios diretos para desenvolver habilidades básicas, grande número de exercícios intermediários e muitos exercícios de desafio. Os exercícios são propostos de maneira clara e sem ambigüidades e todos levam em conta o grau de dificuldade para seu nível. Os conjuntos de exercícios contêm discussões especiais que desenvolvem novos conceitos não abordados no texto, que permitem aos estudantes descobrir novas idéias por meio de seu próprio trabalho.

Exercícios mais difíceis que a maioria estão indicados com um único asterisco *; aqueles que apresentam um grau muito maior de desafio estão indicados com dois asteriscos **. Os exercícios cujas soluções exigem cálculo estão indicados explicitamente. Exercícios que desenvolvem resultados utilizados no texto estão claramente identificados com o símbolo ☞. Respostas ou soluções com explicações de todos os exercícios ímpares encontram-se no final do livro. As respostas incluem demonstrações, nas quais todos os passos são claramente descritos.

QUESTÕES DE REVISÃO Um grupo de questões de revisão está incluso no final de cada capítulo. Essas questões têm como finalidade auxiliar os estudantes em seus estudos dos mais importantes conceitos e técnicas relacionados ao capítulo estudado. Para responder a essas questões, os estudantes precisam escrever longas respostas em vez de apenas apresentar cálculos ou dar respostas curtas.

EXERCÍCIOS COMPLEMENTARES Cada capítulo é acompanhado por um grupo rico e variado de exercícios complementares. Esses exercícios são normalmente mais difíceis que aqueles que acompanham as seções. Os exercícios complementares reforçam os conceitos do capítulo e integram diferentes tópicos de maneira mais eficaz.

DESENVOLVIMENTO DE PROGRAMAS Todos os capítulos são acompanhados de um grupo de projetos computacionais. Os 150 projetos, aproximadamente, interligam o que os estudantes podem aprender na computação e na matemática discreta. Projetos computacionais que são mais difíceis que a maioria, do ponto de vista matemático e de programação, estão indicados com um asterisco, e aqueles que são extremamente desafiadores, com dois asteriscos.

EXPLORANDO A COMPUTAÇÃO Um conjunto de exercícios que envolve computação está incluso no final de cada capítulo. Esses exercícios (em um total de 100, aproximadamente) foram elaborados para ser resolvidos com a ajuda de ferramentas de softwares existentes, como programas que estudantes ou docentes escreveram ou pacotes de computação matemática, como a Maple ou Mathematica. Muitos desses exercícios proporcionam aos estudantes a oportunidade de desvendar novos fatos e idéias no mundo da computação.

PROJETOS Cada capítulo traz alguns projetos para produzir textos. Para esses projetos, os estudantes precisam consultar a literatura matemática. Alguns deles são históricos e podem envolver recursos originais. Outros são desenvolvidos para servir de portais para novos tópicos e idéias. Contudo, todos foram criados para expor idéias aos estudantes, e não abordá-las profundamente no texto. Esses projetos integram conceitos matemáticos ao processo de escrita, ajudando a expor aos estudantes futuras áreas de estudo. (Sugestões de referências para esses projetos podem ser encontradas on-line ou na edição impressa do Guia de Solução do Estudante (*Student's Solutions Guide*). Disponível em inglês.

APÊNDICES Há três apêndices no livro. O primeiro introduz axiomas dos números reais e inteiros e mostra como fatos são demonstrados diretamente a partir desses axiomas. O segundo abrange as funções exponenciais e logarítmicas, revendo o material básico muito usado no curso. O terceiro especifica o pseudocódigo usado para descrever algoritmos no texto.

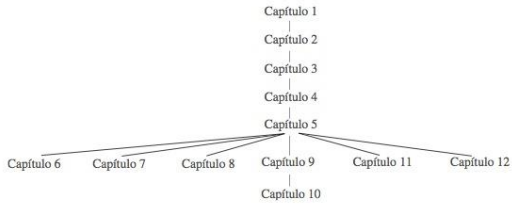
SUGESTÕES DE LEITURA Há, no final do livro, uma seção com uma lista de leituras sugeridas para cada capítulo. Essa lista inclui livros no mesmo nível ou abaixo dele, livros com maior grau de dificuldade, artigos expositivos e artigos originais sobre as descobertas em matemática discreta. Algumas dessas obras são clássicas, publicadas há muitos anos, enquanto outras foram publicadas recentemente.

Como Utilizar Este Livro

Este livro foi cuidadosamente escrito e elaborado a fim de auxiliar nos cursos de matemática discreta em diferentes níveis e com focos diferentes. A tabela a seguir identifica seu conteúdo e as seções opcionais. Um curso introdutório de um semestre em matemática discreta no nível de segundo ano pode se basear nas seções principais do livro, com outras incluídas a partir do interesse do docente. Um curso introdutório de segundo semestre pode incluir todas as seções matemáticas opcionais, além das principais. Um curso com grande ênfase em ciência da computação pode ser ministrado a partir de algumas ou todas as seções opcionais de ciência da computação.

Capítulo	*Seções Centrais*	*Seções Opcionais sobre Ciência da Computação*	*Seções Opcionais sobre Matemática*
1	1.1–1.7 (pré-requisito)		
2	2.1–2.4 (pré-requisito)		
3	3.1–3.5, 3.8 (pré-requisito)	3.6	3.7
4	4.1–4.3	4.4, 4.5	
5	5.1–5.3	5.6	5.4, 5.5
6	6.1	6.4	6.2, 6.3
7	7.1, 7.5	7.3	7.2, 7.4, 7.6
8	8.1, 8.3, 8.5	8.2	8.4, 8.6
9	9.1–9.5		9.6–9.8
10	10.1	10.2, 10.3	10.4, 10.5
11		11.1–11.4	
12		12.1–12.5	

Os professores que forem usar este livro podem ajustá-lo conforme o nível de dificuldade de seu curso, decidindo integrar ou omitir os exemplos mais desafiadores no final das seções, bem como os exercícios mais desafiadores. A dependência dos capítulos é mostrada na figura abaixo.



Materiais Adicionais de Apoio

Este livro conta com um pacote de suplementos para o professor e para o estudante. Esses materiais facilitam o uso do texto e intensificam a experiência de aprendizado, estão disponíveis em inglês e são comerciais, ou seja, você precisa comprá-los. Para adquiri-los, faça um pedido em uma livraria, informando o ISBN.

Guia de Solução do Estudante (*Student's Solutions Guide*) Esse manual contém as resoluções detalhadas de todos os problemas com numeração ímpar. Essas resoluções explicam o porquê do uso de um método em especial e como ele é trabalhado. Para alguns exercícios, uma ou duas outras possibilidades são descritas para mostrar que um problema pode ser resolvido de diversas maneiras. Está também incluso um manual para o desenvolvimento de demonstrações e uma extensa descrição dos erros mais comuns cometidos pelos estudantes em matemática discreta, além de testes amostrais e uma cópia das amostras para cada capítulo, a fim de auxiliar os estudantes em exames.

(ISBN-10: 0-07-310779-4) (ISBN-13: 978-0-07-310779-0)

Guia de Recursos para o Professor (*Instructor's Resource Guide*) Esse manual, disponível a pedido dos professores, contém as respostas completas dos exercícios pares. Sugestões de como usar o material de cada capítulo para o ensino são fornecidas, incluindo em quais pontos deve ser dada ênfase, e como usar o material nesta perspectiva. Testes amostrais para cada capítulo e um banco de exercícios com mais de 1,3 mil questões para escolher, incluindo todas as respostas deste banco e testes, também estão presentes. Por fim, diversas amostras de programas são apresentadas para cursos com diferentes ênfases e níveis de habilidade dos estudantes, e uma seção completa com guia de exercícios de migração foi incluída para ajudar os leitores da quinta edição a atualizar seu material do curso.

(ISBN-10: 0-07-310781-6) (ISBN-13: 978-0-07-310781-3)

CD e outros Recursos para o Professor (*Instructor's Testing and Resource CD*) Um enorme banco de dados com mais de 1,3 mil exercícios, que usa o software Brownstone Diploma, está disponível para os sistemas Windows e Macintosh. Os professores podem usar este software para criar seus próprios testes, selecionando as questões por conta própria ou aleatoriamente. Eles também podem escolher as questões por seção, nível de dificuldade e tipo; editar questões existentes e adicionar suas próprias; incluir suas próprias instruções; imprimir versões do mesmo exercício; exportar exercícios da internet ou de um editor de textos; e criar e gerenciar livros para cursos. Uma versão impressa deste banco de dados, incluindo questões e suas respostas, está inserida no *Guia de Recursos para o Professor.*

(ISBN-10: 0-07-310782-4) (ISBN-13: 978-0-07-310782-0)

Agradecimentos

Gostaria de agradecer aos muitos professores, estudantes e escolas que têm utilizado este livro e têm me auxiliado com suas avaliações e sugestões. Suas interferências fizeram deste um livro muito melhor do que eu esperava. Gostaria de agradecer, em especial, a Jerrold Grossman, John Michaels e George Bergman pela revisão técnica desta sexta edição e pelos seus "olhos de águia", que ajudaram a assegurar a precisão deste livro. Gostaria também de agradecer a todos aqueles que enviaram seus comentários pela internet.

Agradeço aos revisores desta sexta edição e das cinco anteriores. Esses revisores ajudaram com suas críticas construtivas e encorajaram-me. Espero que esta edição supere as suas expectativas.

Revisores da Sexta Edição

Charles Ashbacher,
Mount Mercy College

Ian Barland,
Rice University

George Bergman,
University of California, Berkeley

David Berman,
University of North Carolina, Wilmington

Miklós Bóna,
University of Florida

Prosenjit Bose,
Carleton University

Kirby Brown,
Polytechnic University

Michael Button,
The Master's College

Greg Cameron,
Brigham Young University

John Carter,
University of Toronto

Greg Chapman,
Cosumnes River College

Chao-Kun Cheng,
Virginia Commonwealth University

Thomas Cooper,
Georgia Perimeter College, Lawrenceville

Barbara Cortzen,
DePaul University

Daniel Cunningham,
Buffalo State College

George Davis,
University of Georgia

Beverly Diamond,
College of Charleston

Thomas Dunion,
Atlantic Union College

Bruce Elenbogen,
University of Michigan, Dearborn

Herbert Enderton,
University of California, Los Angeles

Anthony Evans,
Wright State University

Kim Factor,
Marquette University

William Farmer,
McMaster University

Li Feng,
Albany State University

Stephanie Fitchett,
Florida Atlantic University

Jerry Fluharty,
Eastern Shore Community College

Joel Fowler,
Southern Polytechnic State University

William Gasarch,
University of Maryland

Sudpito Ghosh,
Colorado State University

Jerrold Grossman,
Oakland University

Ruth Haas,
Smith College

Patricia Hammer,
Hollins University

Keith Harrow,
Brooklyn College

Bert Hartnell,
St. Mary's University

Julia Hassett,
Oakton College

Kavita Hatwal,
Portland Community College

John Helm,
Radford University

Arthur Hobbs,
Texas A&M University

Fannie Howell,
Roanoke High School

Wei Hu,
Houghton College

Nan Jiang,
University of South Dakota

Margaret Johnson,
Stanford University

Martin Jones,
College of Charleston

Tim Kearns,
California Polytechnic State University, San Luis Obispo

Jonathan Knappenberger,
LaSalle University

Ed Korntved,
Northwest Nazarene University

Przemo Kranz,
University of Mississippi

Loredana Lanzani,
University of Arkansas

Yoonjin Lee,
Smith College

Miguel Lerma,
Northwestern University

Jason Levy,
University of Hawaii

Lauren Lilly,
Tabor College

Ekaterina Lioutikova,
Saint Joseph College

Vladimir Logvinenko,
De Anza College

Joan Lukas,
University of Massachusetts, Boston

Lester McCann,
University of Arizona

Jennifer McNulty,
University of Montana

John G. Michaels,
SUNY Brockport

Michael Oppedisano,
Morrisville State College

Michael O'Sullivan,
San Diego State University

Charles Parry,
Virginia Polytechnic Institute

Linda Powers,
Virginia Polytechnic Institute

Dan Pritikin,
Miami University

Anthony Quas,
University of Memphis

Eric Rawdon,
Duquesne University

Henry Ricardo,
Medgar Evers College

Oskars Rieksts,
Kutztown University

Stefan Robila,
Montclair State University

Chris Rodger,
Auburn University

Robert Rodman,
North Carolina State University

Shai Simonson,
Stonehill College

Barbara Smith,
Cochise College

Wasin So,
San Jose State University

Diana Staats,
Dutchess Community College

Lorna Stewart,
University of Alberta

Bogdan Suceava,
California State University, Fullerton

Kathleen Sullivan,
Seattle University

Laszlo Szekely,
University of South Carolina

Daniel Tauritz,
University of Missouri, Rolla

Don VanderJagt,
Grand Valley State University

Fran Vasko,
Kutztown University

Susan Wallace,
University of North Florida

Zhenyuan Wang,
University of Nebraska, Omaha

Tom Wineinger,
University of Wisconsin, Eau Claire

Charlotte Young,
South Plains College

Revisores das Edições Anteriores

Eric Allender,
Rutgers University

Stephen Andrilli,
La Salle University

Kendall Atkinson,
University of Iowa, Iowa City

Zhaojun Bai,
University of California, Davis

Jack R. Barone,
Baruch College

Klaus Bichteler,
University of Texas, Austin

Alfred E. Borm,
Southwest Texas State University

Ken W. Bosworth,
University of Maryland

Lois Brady,
California Polytechnic State University, San Luis Obispo

Scott Buffett,
University of New Brunswick

Russell Campbell,
University of Northern Iowa

E. Rodney Canfield,
University of Georgia

Kevin Carolan,
Marist College

Tim Carroll,
Bloomsburg University

Kit C. Chan,
Bowling Green State University

Allan C. Cochran,
University of Arkansas

Peter Collinge,
Monroe Community College

Ron Davis,
Millersville University

Nachum Dershowitz,
University of Illinois, Urbana–Champaign

Thomas Dowling,
The Ohio State University

Patrick C. Fischer,
Vanderbilt University

Jane Fritz,
University of New Brunswick

Ladnor Geissinger,
University of North Carolina

Jonathan Goldstine,
Pennsylvania State University

Paul Gormley,
Villanova University

Brian Gray,
Howard Community College

Jonathan Gross,
Columbia University

Laxmi N. Gupta,
Rochester Institute of Technology

Daniel Gusfield,
University of California, Davis

David F. Hayes,
San Jose State University

Xin He,
SUNY at Buffalo

Donald Hutchison,
Clackamas Community College

Kenneth Johnson,
North Dakota State University

David Jonah,
Wayne State University

Akihiro Kanamori,
Boston University

W. Thomas Kiley,
George Mason University

Takashi Kimura,
Boston University

Nancy Kinnersley,
University of Kansas

Gary Klatt,
University of Wisconsin

Nicholas Krier,
Colorado State University

Lawrence S. Kroll,
San Francisco State University

Shui F. Lam,
California State University, Long Beach

Robert Lavelle,
Iona College

Harbir Lamba,
George Mason University

Sheau Dong Lang,
University of Central Florida

Cary Lee,
Grossmont Community College

Yi-Hsin Liu,
University of Nebraska, Omaha

Stephen C. Locke,
Florida Atlantic University

George Luger,
University of New Mexico

David S. McAllister,
North Carolina State University

Robert McGuigan,
Westfield State College

Michael Maller,
Queens College

Ernie Manes,
University of Massachusetts

Francis Masat,
Glassboro State College

J. M. Metzger,
University of North Dakota

Thomas D. Morley,
Georgia Institute of Technology

D. R. Morrison,
University of New Mexico

Ho Kuen Ng,
San Jose State University

Timothy S. Norfolk, Sr.
University of Akron

Truc T. Nguyen,
Bowling Green State University

George Novacky,
University of Pittsburgh

Jeffrey Nunemacher,
Ohio Wesleyan University

Jaroslav Opatrny,
Concordia University

Jonathan Pakianathan,
University of Rochester

Thomas W. Parsons,
Hofstra University

Mary K. Prisco,
University of Wisconsin, Green Bay

Halina Przymusinska,
California Polytechnic State University, Pomona

Don Reichman,
Mercer County Community College

Harold Reiter,
University of North Carolina

Adrian Riskin,
Northern Arizona State University

Amy L. Rocha,
San Jose State University

Janet Roll,
University of Findlay

Matthew J. Saltzman,
Clemson University

Alyssa Sankey,
Slippery Rock University

Dinesh Sarvate,
College of Charleston

Michael J. Schlosser,
The Ohio State University

Steven R. Seidel,
Michigan Technological University

Douglas Shier,
Clemson University

Alistair Sinclair,
University of California, Berkeley

Carl H. Smith,
University of Maryland

Hunter S. Snevily,
University of Idaho

Daniel Somerville,
University of Massachusetts, Boston

Patrick Tantalo,
University of California, Santa Cruz

Bharti Temkin,
Texas Tech University

Wallace Terwilligen,
Bowling Green State University

Philip D. Tiu,
Oglethorpe University

Lisa Townsley-Kulich,
Illinois Benedictine College

George Trapp,
West Virginia University

David S. Tucker,
Midwestern State University

Thomas Upson,
Rochester Institute of Technology

Roman Voronka,
New Jersey Institute of Technology

James Walker,
University of South Carolina

Anthony S. Wojcik,
Michigan State University

Gostaria de agradecer à equipe da McGraw-Hill pelo grande apoio a este projeto. Em particular, agradeço a Liz Haefele, publicitária, por seu auxílio; Liz Covello, editora-chefe, por sua dedicação e entusiasmo; e Dan Seibert, editor de desenvolvimento, por sua dedicação e atenção. Gostaria também de agradecer ao primeiro editor, Wayne Yuhasz, que com suas habilidades e visão ajudou no sucesso do livro, assim como aos editores anteriores deste livro.

Quanto à equipe envolvida na produção desta sexta edição, ofereço meus agradecimentos a Peggy Selle, gerente de projeto; Michelle Whitacker, designer-chefe; Jeff Huettman, produtor de mídia; Sandy Schnee, gerente de projeto de mídia; Melissa Leick, coordenadora de suplementos; e Kelly Brown, gerente de marketing executivo. Sou também grato a Jerry Grossman, que revisou todo o manuscrito; Rose Kramer e Gayle Casel, leitores da prova técnica; Pat Steele, editora do manuscrito; e Georgia Mederer, que checou as respostas do *Guia de Solução do Estudante* e do *Guia de Recursos para o Professor*.

Kenneth H. Rosen

Recursos Disponíveis no Site

O site do livro, em www.mhhe.com/rosen, foi substancialmente melhorado nesta sexta edição e agora está integrado ao MathZone (www.mathzone.com), um sistema de gerenciamento de curso e tutorial on-line. Nele há material para o aluno e para o professor. Esses materiais estão disponíveis em inglês e alguns são comerciais, ou seja, você precisa comprá-los. A página traz links de acesso para ambos, estudante e professor (*Student Edition* e *Instructor Edition*). Para ter acesso aos materiais comerciais, faça um pedido em uma livraria, informando o ISBN do produto. Para ter acesso aos materiais gratuitos, os professores brasileiros necessitam obter uma senha com a McGraw-Hill do Brasil, que deve ser solicitada por e-mail (divulgacao_brasil@mcgraw-hill.com). Na Europa, a senha deve ser obtida com a McGraw-Hill de Portugal (servico_clientes@mcgraw-hill.com).

Por todo o livro aparecerão os ícones a seguir. Estes ícones indicam recursos disponíveis no site que estão integrados ao texto, como exemplos, questões para auto-avaliação, softwares e links úteis. Esses materiais estão disponíveis em inglês.

- **Exemplos Extras** (*Extra Examples*) Você pode encontrar um grande número de exemplos adicionais no site, que abragem todos os capítulos do livro. Esses exemplos estão concentrados em áreas nas quais os estudantes têm mais dúvidas. Embora muitos deles ampliem os conceitos básicos, outros exemplos desafiadores também podem ser encontrados.

- **Demonstração Interativa de Applets** (*Interactive Demonstration Applets*) Esses softwares permitem a você explorar interativamente como os algoritmos importantes funcionam e como

estão ligados diretamente ao conteúdo do texto, assim como aos exemplos e aos exercícios. Recursos adicionais são proporcionados para ajudar em seu uso e sua aplicação.

- **Auto-Avaliação** (*Self Assessments*) Esses guias interativos ajudam você a avaliar seu entendimento dos 14 principais conceitos, por meio de um banco de questões no qual cada questão inclui um pequeno tutorial seguido de uma questão de múltipla escolha. Se você selecionar uma resposta errada, uma dica é dada para ajudá-lo a entender seu erro. Usando este recurso, poderá diagnosticar seus problemas e focalizar as devidas soluções.

- **Guia de Recursos da Internet** (*Web Resources Guide*) Esse guia fornece centenas de links que dão acesso a material relevante, como informações históricas e biográficas, desafios e problemas, discussões, applets, programas, entre outros.

Ao Estudante

O *que é matemática discreta*? Matemática discreta é a parte da matemática voltada para o estudo dos objetos discretos. (Aqui, *discreto* significa elemento distinto ou desconexo). Os tipos de problemas resolvidos com a matemática discreta incluem:

- Quantas formas existem para escolher uma senha válida em um sistema computacional?
- Qual é a probabilidade de se ganhar na loteria?
- Existe um elo entre dois computadores em uma rede de transmissão?
- Como identificar e-mails com *spam*?
- Como posso criptografar uma mensagem para que nenhum destinatário não incluído possa lê-la?
- Qual o menor caminho entre duas cidades usando o sistema de transporte?
- Como se pode sortear uma lista de números inteiros, na qual os números estão em ordem crescente?
- Quantos passos são necessários para se fazer uma escolha?
- Como é possível provar que um algoritmo escolhido corretamente resolve determinado problema?
- Como é determinado um circuito que soma dois inteiros?
- Quantos endereços válidos da internet existem?

Você vai aprender as estruturas e as técnicas discretas necessárias para resolver problemas como esses.

Generalizando, a matemática discreta é usada quando os objetos são contados, quando são estudadas as relações entre os conjuntos finitos (ou contáveis) e quando são analisados os processos que envolvem um número finito de passos. A principal razão para o crescimento da importância da matemática discreta é que a informação é administrada e manipulada por computadores de uma maneira discreta.

POR QUE ESTUDAR MATEMÁTICA DISCRETA? Há muitas razões importantes para se estudar a matemática discreta. Em primeiro lugar, ao longo deste curso você poderá desenvolver sua maturidade matemática, ou seja, sua capacidade de entender e criar argumentos matemáticos. Você não conseguirá avançar muito em seus estudos matemáticos sem essas habilidades.

Em segundo lugar, a matemática discreta é a porta de entrada para cursos mais avançados de todas as áreas das ciências matemáticas. Ela dá as bases matemáticas para muitos cursos de ciência da computação, incluindo estruturas de dados, algoritmos, teoria da base de dados, teoria da automação, linguagens formais, teoria da compilação, segurança computacional e sistemas de operações. Os estudantes encontram muito mais dificuldade nesses estudos quando não possuem os fundamentos básicos apropriados da matemática discreta. Uma estudante enviou-me um e-mail dizendo que costuma usar o conteúdo deste livro em todos os cursos de ciência da computação que ela freqüenta!

Cursos de matemática que se baseiam no conteúdo estudado pela matemática discreta incluem lógica, teoria dos conjuntos, teoria dos números, álgebra linear, álgebra abstrata, análise combinatória, teoria dos grafos e teoria da probabilidade (a parte discreta de cada tópico).

Além disso, a matemática discreta contém o contexto matemático necessário para resolver problemas em pesquisas operacionais (incluindo muitas técnicas de otimização discreta), química, engenharia e biologia, entre outros. Nesta obra, estudaremos algumas aplicações nessas áreas.

Muitos estudantes julgam que o curso de introdução à matemática discreta seja significativamente mais desafiante que os cursos que já tiveram. Uma razão para isso é que o objetivo inicial deste curso é ensinar a argumentação matemática e a resolução de problemas, em vez das habilidades dos conjuntos discretos. Os exercícios neste livro são projetados para refletir esse objetivo. Embora haja diversos exercícios nesta obra similares aos tratados nos exemplos, uma grande

porcentagem deles requer pensamento original. Isso é intencional, o conteúdo discutido no texto proporciona as ferramentas necessárias para a resolução desses exercícios, mas seu trabalho é aplicar, com sucesso, essas ferramentas com sua própria criatividade. Um dos principais objetivos deste curso é aprender a lidar com problemas que podem ser diferentes daqueles que você viu e resolveu anteriormente. Infelizmente, aprender a solucionar um tipo particular de exercício não é suficiente para ter sucesso no desenvolvimento das habilidades necessárias para a resolução de problemas nos cursos subseqüentes e no trabalho profissional. Este texto é direcionado a muitos tópicos diferentes, uma vez que a matemática discreta é uma área do conhecimento extremamente ampla e diversificada. Um dos objetivos como autor é ajudá-lo a desenvolver as habilidades necessárias para maximizar as informações adicionais de que você necessitará em sua caminhada futura.

OS EXERCÍCIOS Gostaria de dar alguns conselhos sobre como você pode aprender melhor a matemática discreta (e outros assuntos nas ciências matemática e computacional). Você aprenderá ao máximo a partir da resolução dos exercícios. Sugiro que você resolva o máximo de exercícios possíveis. Depois de trabalhar os exercícios dados por seu professor, eu o encorajo a resolver os exercícios adicionais, assim como aqueles que acompanham cada seção no texto e os exercícios complementares, no final de cada capítulo (veja a nota que explica as marcações que acompanham os exercícios).

Nota aos Exercícios

Não marcados	Um exercício de rotina
*	Um exercício difícil
**	Um exercício extremamente desafiador
☞	Um exercício que contém um resultado utilizado no livro (A Tabela abaixo mostra onde cada um desses exercícios é usado.)
(*Requer cálculo*)	Um exercício que, em sua solução, exige o uso de limites ou conceitos de cálculos diferenciais ou integral.

Exercícios com o ícone "mão" e onde eles são usados

Seção	*Exercício*	*Seção onde são usados*	*Páginas onde são usados*
1.2	42	11.2	758
1.6	16	1.6	81
2.3	75a	2.4	158
3.1	43	3.1	174
3.2	62	10.2	700
3.7	30	6.2	412
4.1	79	4.1	270
4.2	28	4.2	287
4.3	56	4.3	296
4.3	57	4.3	297
5.4	17	6.2	414
5.4	21	6.4	428
6.2	15	6.2	414
8.1	24	8.4	545
9.4	49	10.1	685
10.1	15	10.1	688
10.1	30	10.1	693
10.1	48	10.2	700
11.1	12	11.3	763
A.2	4	7.3	477

A melhor forma de estudo é tentar resolver os exercícios sozinho antes de consultar as respostas no final do livro. Note que os exercícios ímpares possuem apenas as respostas, e não as soluções completas; normalmente, a argumentação necessária para obter as respostas é omitida. O Guia de Solução do Estudante (*Student's Solutions Guide*) proporciona as soluções completas para todos os exercícios ímpares. O *Student's Solutions Guide* é um produto comercial e está disponível em inglês, consulte a seção *Materiais Adicionais de Apoio*, no Prefácio, para obter mais informações. Ao se deparar com um impasse para resolver um problema, é recomendável a consulta a esse material ou a busca de ajuda para resolver a questão. Quanto mais exercícios você resolver sozinho, sem a cópia passiva das respostas, mais você aprenderá. As respostas e as soluções dos exercícios pares foram omitidas intencionalmente na publicação; pergunte a seu professor se você tiver problemas com eles.

RECURSOS NA INTERNET Você está sendo extremamente encorajado a aproveitar os recursos adicionais disponíveis no site do livro em www.mhhe.com/rosen, no link *Student Edition*. Você encontrará muitos exemplos extras criados para elucidar os principais conceitos; auto-avaliação para verificar como está entendendo o conteúdo dos tópicos; demonstração interativa de Applets que exploram os principais algoritmos e outros conceitos; um guia de recursos da internet que contém uma seleção extensa dos links de sites interessantes do mundo da matemática discreta; explicações e prática extras para ajudá-lo a dominar ao máximo os conceitos; instruções adicionais para o desenvolvimento de demonstrações e como evitar erros comuns em matemática discreta; discussões aprofundadas em importantes aplicações e um guia de como utilizar o software da "Maple" para explorar os aspectos computacionais da matemática discreta. Lugares no texto onde estão disponíveis estes recursos *on-line* estão identificados por ícones especiais nas margens. Todos esses recursos estão disponíveis em inglês.

O VALOR DESTE LIVRO Minha intenção é fazer do seu investimento neste livro um excelente negócio. O livro e os materiais de apoio consumiram muitos anos de esforço para seu desenvolvimento e refinamento. Estou certo de que a maioria achará que o texto e os materiais extras ajudarão no domínio da matemática discreta. Mesmo se alguns capítulos não forem abordados ao longo do curso, você descobrirá que será uma ótima ajuda — assim como muitos estudantes descobriram — ler seções relevantes do livro. A maioria terá neste livro uma ferramenta útil aos estudos futuros, especialmente aqueles que continuarem na ciência da computação, na matemática e na engenharia. Eu criei este livro para ser a porta de entrada para estudos e explorações futuras e desejo-lhe sorte ao começar sua jornada.

Kenneth H. Rosen

CAPÍTULO

1 Os Fundamentos: Lógica e Demonstrações

1.1 Lógica Proposicional
1.2 Equivalências Proposicionais
1.3 Predicados e Quantificadores
1.4 Quantificadores Agrupados
1.5 Regras de Inferência
1.6 Introdução a Demonstrações
1.7 Métodos de Demonstração e Estratégia

As regras da lógica especificam o significado de sentenças matemáticas. Propositalmente, essas regras nos ajudam a entender proposições tais como "Existe um inteiro que não é a soma de dois quadrados" e "Para cada inteiro positivo n, a soma dos inteiros positivos menores que ou iguais a n é $n(n + 1) / 2$", bem como a raciocinar sobre elas. Lógica é a base de todo raciocínio matemático e de todo raciocínio automatizado. Ela tem aplicações práticas no desenvolvimento de máquinas de computação, em especificação de sistemas, em inteligência artificial, em programação de computadores, em linguagens de programação e em outras áreas da ciência da computação, bem como em outros campos de estudo.

Para entender matemática, precisamos entender o que torna um argumento matemático correto, ou seja, uma demonstração. Primeiro, demonstramos que uma afirmação é verdadeira e a chamamos de teorema. Um conjunto de teoremas sobre determinado tópico organiza o que conhecemos sobre esse tópico. Para aprender um tópico de matemática, uma pessoa precisa construir ativamente argumentos matemáticos nesse tópico, e não apenas ler algumas exposições. Além disso, conhecer a demonstração de um teorema, com freqüência, torna possível modificar o resultado em alguma outra situação; demonstrações são essenciais para o desenvolvimento de novas idéias. Estudantes de ciência da computação freqüentemente se surpreendem em relação ao quanto as demonstrações são importantes em sua área. De fato, elas são essenciais quando queremos verificar se um programa computacional está dando uma saída correta para todos os possíveis valores de entrada, quando mostramos que algoritmos sempre produzem resultados corretos, quando estabelecemos a segurança de um sistema e quando criamos inteligência artificial. Sistemas de raciocínio automatizado têm sido construídos para permitir que computadores construam suas próprias demonstrações.

Neste capítulo, vamos explicar o que torna um argumento matemático correto e introduzir ferramentas para construir esses argumentos. Vamos desenvolver um arsenal de métodos de demonstração diferentes que nos tornarão capazes de demonstrar muitos tipos de resultados. Depois de introduzir os diferentes métodos de demonstração, introduziremos alguma estratégia para a construção de demonstrações e também a noção de conjectura, e explicaremos o processo de desenvolvimento da matemática pelo estudo das conjecturas.

1.1 Lógica Proposicional

Introdução

As regras de lógica nos dão um significado preciso para sentenças matemáticas. Essas regras são usadas para distinguir entre argumentos matemáticos válidos e inválidos. Como o objetivo principal deste livro é ensinar nosso leitor a entender e a construir argumentos matemáticos corretos, começaremos nosso estudo de matemática discreta com uma introdução à lógica.

Paralelamente à sua importância no entendimento do raciocínio matemático, a lógica tem numerosas aplicações em ciência da computação. Suas regras são usadas no design de circuitos de computador, na construção de programas, na verificação da correção de programas e de muitas outras formas. Além do mais, sistemas de softwares têm sido desenvolvidos para construir demonstrações automaticamente. Vamos discutir essas aplicações da lógica nos próximos capítulos.

Proposições

Nossa discussão começa com uma introdução à construção dos blocos básicos de lógica — proposições. Uma **proposição** é uma sentença declarativa (isto é, uma sentença que declara um fato), que pode ser verdadeira ou falsa, mas não ambas.

EXEMPLO 1 Todas as seguintes sentenças declarativas são proposições.

Exemplos Extras

1. Brasília é a capital do Brasil.
2. Toronto é a capital do Canadá.
3. $1 + 1 = 2$.
4. $2 + 2 = 3$.

As proposições 1 e 3 são verdadeiras, assim como 2 e 4 são falsas. ◄

Algumas sentenças que não são proposições são dadas no Exemplo 2.

EXEMPLO 2 Considere as seguintes sentenças.

1. Que horas são?
2. Leia isto cuidadosamente.
3. $x + 1 = 2$.
4. $x + y = z$.

As sentenças 1 e 2 não são proposições porque não são sentenças declarativas. As sentenças 3 e 4 não são proposições porque não são nem verdadeiras nem falsas. Note que as sentenças 3 e 4 podem se tornar proposições se atribuirmos um valor para as variáveis. Também vamos discutir outros meios de transformar tais sentenças em proposições na Seção 1.3. ◄

Usamos letras para indicar **variáveis proposicionais (ou variáveis declarativas)**, que são variáveis que representam proposições, assim como letras são usadas para indicar variáveis numéricas. As letras convencionalmente usadas para variáveis proposicionais são *p, q, r, s,....* O **valor-verdade** de uma proposição é verdadeiro, indicado por V, se for uma proposição verdadeira, e falso, indicado por F, se for uma proposição falsa.

A área da lógica que se preocupa com as proposições é chamada de **cálculo proposicional** ou **lógica proposicional**, e foi desenvolvida sistematicamente a primeira vez pelo filósofo grego Aristóteles, há mais de 2.300 anos.

Links

ARISTÓTELES (384 a.C.–322 a.C.) Aristóteles nasceu em Estagira, norte da Grécia. Seu pai era o médico particular do rei da Macedônia. Como seu pai morreu quando Aristóteles era jovem, o filósofo não pôde seguir o costume de ter a mesma profissão de seu pai. Aristóteles ficou órfão cedo, pois sua mãe também morreu logo. O guardião que o criou ensinou-lhe poesia, retórica e grego. Aos 17 anos, seu guardião o enviou a Atenas para continuar sua educação. Aristóteles juntou-se à Academia de Platão, onde freqüentou durante 20 anos as aulas de Platão, e posteriormente apresentou suas próprias leituras de retórica. Quando Platão morreu em 347 a.C., Aristóteles não foi escolhido para ser seu sucessor na Academia, pois suas idéias eram muito diferentes das de Platão. Assim, Aristóteles se juntou à corte do rei Hérmias, onde permaneceu por três anos e casou-se com a sobrinha do rei. Como os persas derrotaram Hérmias, Aristóteles se mudou para Mitilena e, convidado pelo rei Filipe da Macedônia, tornou-se tutor de Alexandre, filho de Filipe, que mais tarde se tornaria Alexandre, o Grande. Aristóteles foi tutor de Alexandre durante cinco anos e, depois da morte do rei Filipe, retornou a Atenas e organizou sua própria escola, chamada "Liceu".

Os seguidores de Aristóteles eram chamados de "peripatéticos", que significa "os que passeiam", já que Aristóteles costumava caminhar quando discutia questões filosóficas. Aristóteles ensinou no "Liceu" por 13 anos, dando lições aos seus estudantes mais avançados pela manhã e aos populares em um auditório aberto, à noite. Quando Alexandre, o Grande, morreu em 323 a.C., uma reação violenta contra tudo relacionado a Alexandre iniciou um grande ataque de ímpetos contra Aristóteles, que fugiu para o Cálcis para evitar um processo. Ele viveu apenas um ano em Cálcis, morrendo por um problema estomacal em 322 a.C.

Aristóteles escreveu três tipos de trabalho: aqueles escritos para uma audiência popular, compilações de fatos científicos e tratados sistemáticos. Estes últimos incluem trabalhos de lógica, filosofia, psicologia, física e história natural. Os escritos de Aristóteles foram preservados por um estudante e escondidos em uma cripta, sendo descobertos por um colecionador de livros 200 anos depois. Eles foram levados para Roma, onde foram estudados por eruditos e publicados em novas edições para serem preservados para a posteridade.

Links

Agora voltaremos nossa atenção para métodos de produção de novas proposições a partir daquelas que já temos. Esses métodos foram discutidos pelo matemático inglês George Boole em 1854, em seu livro *The Laws of Thought*. Muitas sentenças matemáticas são construídas combinando-se uma ou mais proposições. Novas proposições, chamadas de **proposições compostas**, são criadas a partir de proposições existentes usando-se operadores lógicos.

DEFINIÇÃO 1 Seja p uma proposição. A *negação de p*, indicada por $\neg p$ (e também por $\overline{p}$), é a sentença

"Não é o caso de p."

A proposição $\neg p$ é lida "não p". O valor-verdade da negação de p, $\neg p$, é o oposto do valor-verdade de p.

EXEMPLO 3 Encontre a negação da proposição

"Hoje é sexta-feira."

Exemplos Extras

e expresse-a em português.

Solução: A negação é

"Não é o caso de hoje ser sexta-feira."

A negação pode ser expressa simplesmente por

"Hoje não é sexta-feira."

ou

"Não é sexta-feira hoje." ◀

EXEMPLO 4 Encontre a negação da proposição

"No mínimo 10 mm de chuva caíram hoje em São Paulo."

e expresse-a em português.

Solução: A negação é

"Não é o caso de no mínimo 10 mm de chuva ter caído hoje em São Paulo."

E pode ser expressa por

"Menos de 10 mm de chuva caíram hoje em São Paulo." ◀

Lembre-se: No sentido estrito, sentenças que envolvem advérbios de tempo como essas dos exemplos 3 e 4 não são proposições até que um tempo fixo seja assumido. O mesmo ocorre para advérbios de lugar, até que um lugar fixo seja assumido, e para pronomes, até que um indivíduo seja assumido. Nós sempre assumiremos um instante fixo, um lugar definido ou um indivíduo determinado nessas sentenças, a não ser que indiquemos o contrário.

TABELA 1 A Tabela-Verdade para a Negação de uma Proposição.

p	$\neg p$
V	F
F	V

A Tabela 1 nos mostra a **tabela-verdade** para a negação de uma proposição p. Essa tabela tem uma linha para cada uma das possibilidades de valor-verdade da proposição p – Verdadeiro e Falso. Cada linha mostra o valor-verdade de $\neg p$ correspondente ao valor-verdade de p nesta linha.

A negação de uma proposição pode também ser considerada o resultado da operação do **operador de negação** em uma proposição. O operador de negação constrói novas proposições a partir de proposições preexistentes. Agora, introduziremos os operadores lógicos que são usados para criar novas proposições a partir de outras duas ou mais já existentes. Os operadores lógicos são também chamados de **conectivos**.

DEFINIÇÃO 2 Sejam p e q proposições. A *conjunção* de p e q, indicada por $p \wedge q$, é a proposição "p e q". A conjunção $p \wedge q$ é verdadeira quando ambas são verdadeiras, e falsa caso contrário.

A Tabela 2 nos mostra a tabela-verdade para $p \wedge q$. Essa tabela tem uma linha para cada combinação de valores-verdade para as proposições p e q. As quatro linhas correspondem aos pares VV, VF, FV e FF, em que o primeiro valor-verdade é o valor de p e o segundo é o valor de q.

Note que em lógica a palavra "mas" é freqüentemente usada no lugar do "e" em uma conjunção. Por exemplo, a frase "O sol está brilhando, mas está chovendo" é uma outra maneira de dizer "O sol está brilhando e está chovendo". (Em nossa linguagem natural, existe uma diferença substancial entre "mas" e "e", mas não vamos nos preocupar com essa nuança aqui.)

EXEMPLO 5 Encontre a conjunção das proposições p e q, em que p é a proposição "Hoje é sexta-feira" e q é a proposição "Hoje está chovendo".

Solução: A conjunção $p \wedge q$ dessas proposições é a proposição "Hoje é sexta-feira e hoje está chovendo". Essa proposição é verdadeira em uma sexta-feira chuvosa e é falsa em qualquer outro caso. ◀

DEFINIÇÃO 3 Sejam p e q proposições. A *disjunção* de p e q, indicada por $p \vee q$, é a proposição "p ou q". A disjunção $p \vee q$ é falsa se p e q são ambas falsas, e verdadeira em qualquer outro caso.

A Tabela 3 nos mostra a tabela-verdade para $p \vee q$.

O uso do conectivo *ou* em uma disjunção corresponde a uma das formas de uso da palavra *ou* em português, chamado *ou inclusivo*. Uma disjunção é verdadeira quando ao menos uma das proposições é verdadeira. Por exemplo, o ou inclusivo foi usado na frase

> "Estudantes que têm aulas de cálculo ou ciência da computação podem assistir a estas aulas."

TABELA 2 A Tabela-Verdade para a Conjunção de Duas Proposições.

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

TABELA 3 A Tabela-Verdade para a Disjunção de Duas Proposições.

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

Aqui, queremos dizer que estudantes que têm aulas de cálculo e ciência da computação podem assistir a estas aulas, bem como estudantes que têm aulas em apenas um dos cursos. Por outro lado, se queremos usar o *ou exclusivo*, dizemos

> "Estudantes que têm aulas de cálculo ou ciência da computação, mas não ambas, podem assistir a estas aulas."

Nesse caso, queremos dizer que estudantes que têm aulas nesses dois cursos, cálculo e ciência da computação, não podem assistir a estas aulas.

De maneira similar, quando em um cardápio de restaurante está escrito "Sopa ou salada é servida como entrada", o restaurante quer dizer que uma das duas entradas pode ser pedida, mas não ambas. Portanto, esse é um ou exclusivo, e não um ou inclusivo.

EXEMPLO 6 Qual é a disjunção das proposições p e q, em que p e q são as proposições do Exemplo 5?

Exemplos Extras

Solução: A disjunção $p \vee q$ dessas proposições é a proposição

"Hoje é sexta-feira ou hoje está chovendo."

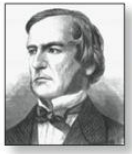
Essa proposição é verdadeira em uma sexta-feira ou em um dia chuvoso e somente é falsa no caso em que não é sexta-feira nem chove. ◀

Como tínhamos observado anteriormente, o uso do conectivo *ou* em uma disjunção corresponde a uma das formas da palavra *ou* usadas em português, denominada forma inclusiva. Então, uma disjunção é verdadeira se uma das proposições o é. Às vezes usamos *ou* da maneira exclusiva. Quando a maneira exclusiva é usada para conectar as proposições p e q, a proposição "*p ou q (mas não ambas)*" é obtida. Essa proposição é verdadeira se uma, e apenas uma, das proposições é verdadeira, e é falsa quando p e q são ambas falsas ou ambas verdadeiras.

DEFINIÇÃO 4 Sejam p e q proposições. A *disjunção exclusiva* (ou *ou exclusivo*) de p e q, indicada por $p \oplus q$, é a proposição que é verdadeira quando exatamente uma das duas é verdadeira e falsa nos outros casos.

A tabela-verdade para o ou exclusivo de duas proposições é mostrada na Tabela 4.

Links



GEORGE BOOLE (1815–1864) George Boole, filho de um sapateiro, nasceu em Lincoln, na Inglaterra, em novembro de 1815. Por causa da difícil situação financeira da família, Boole teve que se esforçar para educar-se enquanto a sustentava. No entanto, ele se tornou um dos matemáticos mais importantes do século XIX. Embora considerasse a carreira como um clérigo, ele preferiu, em vez disso, entrar no mundo do ensino e mais tarde abrir sua própria escola. Em sua preparação para ensinar matemática, Boole — insatisfeito com os livros de sua época — decidiu ler os trabalhos dos grandes matemáticos. Enquanto lia os trabalhos do grande matemático francês Lagrange, Boole fez descobertas no cálculo de variantes, o campo de análise que lida com a descoberta de curvas e superfícies e, assim, otimiza certos parâmetros.

Em 1848, Boole publicou o livro *The Mathematical Analysis of Logic*, o primeiro de sua contribuição à lógica simbólica. Em 1849, ele foi convidado para ser professor de matemática da Universidade de Queen, em Cork, Irlanda. Em 1854, publicou *The Laws of Thought*, seu mais famoso trabalho. Nesse livro, Boole introduziu o que passou a ser chamado de *álgebra booleana*, em sua homenagem. Boole escreveu livros de teoria de equações diferenciais que foram usados na Grã-Bretanha até o final do século XIX. Casou-se em 1855; sua mulher era a sobrinha do professor de grego da Universidade de Queen. Em 1864, Boole morreu de pneumonia, contraída por manter-se lendo mesmo encharcado depois de uma tempestade.

TABELA 4 A Tabela-Verdade para o Ou Exclusivo de Duas Proposições.

p	q	$p \oplus q$
V	V	F
V	F	V
F	V	V
F	F	F

TABELA 5 A Tabela-Verdade para as Sentenças Condicionais $p \rightarrow q$.

p	q	$p \rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

Proposições Condicionais

Vamos, agora, discutir outros modos importantes sobre os quais podemos combinar proposições.

DEFINIÇÃO 5

Sejam p e q proposições. A *proposição condicional* $p \rightarrow q$ é a proposição "se p, então q". A condicional $p \rightarrow q$ é falsa quando p é verdadeira e q é falsa, e verdadeira em qualquer outro caso. Na condicional $p \rightarrow q$, p é chamada de *hipótese* (ou *antecedente* ou *premissa*) e q é chamada de *conclusão* (ou *conseqüência* ou *conseqüente*).

Auto-avaliação

A proposição $p \rightarrow q$ é chamada de condicional porque $p \rightarrow q$ afirma que q é verdadeira na condição de que p também o seja. Uma proposição condicional é também chamada de **implicação**.

A tabela-verdade para a condicional $p \rightarrow q$ é mostrada na Tabela 5. Note que $p \rightarrow q$ é verdadeira quando ambos o são e quando p é falsa (não importando qual o valor-verdade de q).

Como a condicional é usada como uma regra essencial no raciocínio matemático, uma variedade de termos pode ser usada para expressar $p \rightarrow q$. Você pode encontrar algumas das seguintes formas para expressar a condicional:

"se p, então q"
"se p, q"
"p é suficiente para q"
"q se p"
"q quando ocorrer p"
"uma condição necessária para p é q"
"q a menos que $\neg p$"
"p implica q"
"p apenas se q"
"uma condição suficiente para q é p"
"q sempre que p"
"q é necessário para p"
"q segue de p"

Uma maneira usual de entender a tabela-verdade de uma condicional é pensar em uma obrigação ou um contrato. Por exemplo, uma promessa que muitos políticos fazem quando são candidatos é

"Se eu for eleito, então vou diminuir os impostos."

Se o político for eleito, os eleitores devem esperar que esse político diminua os impostos. No entanto, se o político não for eleito, os eleitores não terão nenhuma expectativa sobre o que tal político fará com os impostos, mesmo que a pessoa tenha influência suficiente para baixá-los. Será apenas quando o político for eleito, mas não baixar os impostos, que os eleitores poderão dizer que o político quebrou sua promessa de campanha. Esse último cenário corresponde ao caso em que p é verdadeira e q é falsa em $p \rightarrow q$.

Similarmente, considere a proposição que um professor pode fazer:

"Se você tirar nota 10 no exame final, então terá conceito A."

Se tirar nota 10 no exame final, então você espera receber o conceito A. Se não tirar 10, você pode ou não receber o conceito A dependendo de outros fatores. No entanto, se tirar 10, mas o professor não lhe der o conceito A, então você se sentirá trapaceado.

Muitas pessoas acham confuso que "p somente se q" expresse o mesmo que "se p então q". Para lembrar isto, note que "p somente se q" significa que p não pode ser verdadeira quando q não é. Ou seja, a proposição é falsa se p é verdadeira, mas q é falsa. Quando p é falsa, q pode ser verdadeira ou falsa, porque a proposição não diz nada sobre o valor-verdade de q. Um erro comum é pensar que "q somente se p" é uma possibilidade de expressar $p \rightarrow q$. No entanto, essas proposições têm valores-verdade diferentes quando p e q têm valores-verdade diferentes.

A expressão "a menos que" é freqüentemente usada para expressar condicionais. Observe que "q a menos que $\neg p$" significa que se $\neg p$ é falsa, então q deve ser verdadeira. Ou seja, a proposição "q a menos que $\neg p$" é falsa quando p é verdadeira e q é falsa, mas é verdadeira em qualquer outro caso. Conseqüentemente, "q a menos que $\neg p$" e $p \rightarrow q$ têm o mesmo valor-verdade.

Ilustraremos a tradução entre condicionais e proposições em português no Exemplo 7.

EXEMPLO 7 Seja p a proposição "Maria aprende matemática discreta" e q a proposição "Maria vai conseguir um bom emprego". Expresse $p \rightarrow q$ em português.

Exemplos Extras

Solução: Da definição de condicional, vemos que, quando p é a proposição "Maria aprende matemática discreta" e q é a proposição "Maria vai conseguir um bom emprego", $p \rightarrow q$ representa a proposição

"Se Maria aprender matemática discreta, então ela vai conseguir um bom emprego."

Existem muitas outras maneiras de expressar essa condicional em português. Algumas das mais naturais são:

"Maria vai encontrar um bom emprego quando aprender matemática discreta."

"Para conseguir um bom emprego, é suficiente que Maria aprenda matemática discreta."

e

"Maria vai conseguir um bom emprego, a menos que não aprenda matemática discreta." ◀

Note que o modo que definimos a condicional é mais geral do que o significado intrínseco às proposições em português. Veja que a proposição condicional no Exemplo 7 e a proposição

"Se hoje está ensolarado, então vou à praia."

são proposições utilizadas em linguagem natural, em que há uma relação entre a hipótese e a conclusão. Além disso, a primeira proposição é verdadeira, a menos que Maria aprenda matemática discreta, mas não consiga um bom emprego, e a segunda é verdadeira, a menos que seja um dia ensolarado e eu não vá à praia. Por outro lado, a proposição

"Se hoje é sexta-feira, então $2 + 3 = 5$."

é verdadeira pela definição de proposição condicional, porque a conclusão é verdadeira. (O valor-verdade da hipótese não faz diferença nesse caso.) A proposição condicional

"Se hoje é sexta-feira, então $2 + 3 = 6$."

é verdadeira para todos os dias, exceto às sextas-feiras, já que $2 + 3 = 6$ é falso.

Não usamos essas duas últimas condicionais em linguagem natural (exceto sarcasticamente), porque não existe uma relação entre a hipótese e a conclusão nos dois casos. No raciocínio

matemático, consideramos a condicional de uma forma mais geral que a usada na língua portuguesa. O conceito matemático de condicional é independente de relações de causa–efeito entre a hipótese e a conclusão. Nossa definição de condicional especifica seus valores-verdade; não tem como base a sua utilização em português. A linguagem proposicional é uma linguagem artificial; nós apenas usamos o paralelo em português para torná-la fácil de ser utilizada e lembrada.

A construção se-então usada em muitas linguagens de programação é diferente da usada em lógica. A maioria das linguagens de programação contém declarações tais como **if** p **then** S, em que p é uma proposição e S é um segmento do programa (uma ou mais declarações a serem executadas). Quando a execução do programa encontra tal declaração, S é executado se p é verdadeira, mas S não é executado se p é falsa, como ilustrado no Exemplo 8.

EXEMPLO 8 Qual o valor da variável x depois da declaração

if $2 + 2 = 4$ **then** $x := x + 1$

se $x = 0$ antes de a declaração ser encontrada? (O símbolo $:=$ significa atribuição. A declaração $x := x + 1$ significa que o valor $x + 1$ será atribuído a x.)

Solução: Como $2 + 2 = 4$ é verdadeira, a declaração de atribuição $x := x + 1$ é executada. Portanto, x tem o valor $0 + 1 = 1$ depois da declaração encontrada. ◀

OPOSTA, CONTRAPOSITIVA E INVERSA Podemos formar muitas outras proposições começando com a condicional $p \rightarrow q$. Em particular, existem três proposições condicionais relacionadas que ocorrem tão freqüentemente que damos nomes a elas. A proposição $q \rightarrow p$ é chamada de **oposta** de $p \rightarrow q$. A **contrapositiva** de $p \rightarrow q$ é a proposição $\neg q \rightarrow \neg p$. A proposição $\neg p \rightarrow \neg q$ é chamada de **inversa** de $p \rightarrow q$. Veremos que, dessas três condicionais formadas a partir de $p \rightarrow q$, apenas a contrapositiva sempre tem o mesmo valor-verdade que $p \rightarrow q$.

Primeiro vamos mostrar que a contrapositiva, $\neg q \rightarrow \neg p$, de uma condicional $p \rightarrow q$ sempre tem o mesmo valor-verdade que $p \rightarrow q$. Para ver isso, note que a contrapositiva é falsa apenas no caso de $\neg p$ falsa e $\neg q$ verdadeira, que é apenas quando q é falsa e p é verdadeira. Agora podemos ver que nem a oposta, $q \rightarrow p$, nem a inversa, $\neg p \rightarrow \neg q$, têm o mesmo valor-verdade que $p \rightarrow q$, para todos os possíveis valores-verdade de p e q. Note que, quando p é verdadeira e q é falsa, a condicional original é falsa, mas a oposta e a inversa são ambas verdadeiras.

Quando duas proposições compostas têm sempre o mesmo valor-verdade, nós as chamamos **equivalentes**, de modo que a proposição condicional e a contrapositiva são equivalentes. A oposta e a inversa também são proposições equivalentes, como o leitor pode verificar, mas não são equivalentes à condicional original. (Vamos estudar proposições equivalentes na Seção 1.2.) Considere nota que um dos erros mais comuns na lógica é assumir que a inversa ou a oposta de uma condicional é equivalente a essa condicional.

Ilustraremos o uso das proposições condicionais no Exemplo 9.

EXEMPLO 9 Qual é a contrapositiva, a oposta e a inversa da proposição condicional

"O time da casa ganha sempre que está chovendo."?

Exemplos Extras

Solução: Como "q sempre que p" é uma das maneiras de escrever a condicional $p \rightarrow q$, a proposição original pode ser reescrita como

"Se está chovendo, então o time da casa ganha."

Conseqüentemente, a contrapositiva dessa condicional é

"Se o time da casa não ganha, então não está chovendo."

A oposta é

"Se o time da casa ganha, então está chovendo."

A inversa é

"Se não está chovendo, então o time da casa não ganha."

Apenas a contrapositiva é equivalente à condicional original. ◀

BICONDICIONAIS Vamos agora introduzir uma nova maneira de combinar proposições que expressa que duas proposições têm o mesmo valor-verdade.

DEFINIÇÃO 6 Sejam p e q proposições. A *proposição bicondicional* $p \leftrightarrow q$ é a proposição "p se e somente se q". A bicondicional $p \leftrightarrow q$ é verdadeira sempre que p e q têm o mesmo valor-verdade, e falsa caso contrário. Bicondicionais são também chamadas de *bi-implicações*.

A tabela-verdade para $p \leftrightarrow q$ está representada na Tabela 6. Note que $p \leftrightarrow q$ é verdadeira, quando ambas as condicionais $p \rightarrow q$ e $q \rightarrow p$ são verdadeiras, e falsa, caso contrário. Este é o motivo pelo qual usamos a expressão "se e somente se" para expressar o conectivo e porque ele é simbolicamente escrito pela combinação de $\rightarrow$ e $\leftarrow$. Existem muitas outras maneiras comuns de expressar $p \leftrightarrow q$:

"p é necessária e suficiente para q"
"se p então q, e vice-versa"
"p sse q" ("p iff q").

A última maneira de expressar uma bicondicional usa a abreviação "sse" para "se e somente se". Note que $p \leftrightarrow q$ é exatamente o mesmo que $(p \rightarrow q) \wedge (q \rightarrow p)$.

EXEMPLO 10 Seja p a proposição "Você pode tomar o avião" e q a proposição "Você comprou uma passagem". Então $p \leftrightarrow q$ é a proposição

"Você pode tomar o avião se e somente se você comprou uma passagem."

Exemplos Extras Essa proposição é verdadeira se p e q são ambas verdadeiras ou ambas falsas, ou seja, se você comprou uma passagem e pode tomar o avião ou se você não comprou uma passagem e não pode tomar o avião. E ela é falsa quando p e q têm valores-verdade opostos, ou seja, quando você comprou uma passagem, mas não pode tomar o avião (como se a empresa aérea o impedisse) ou quando não comprou uma passagem e pode tomar o avião (como se você tivesse ganho uma viagem). ◀

TABELA 6 A Tabela-Verdade para a Bicondicional $p \leftrightarrow q$.

p	q	$p \leftrightarrow q$
V	V	V
V	F	F
F	V	F
F	F	V

USO IMPLÍCITO DE BICONDICIONAIS Devemos estar cientes de que nem sempre bicondicionais estão explícitas em nossa linguagem natural. Em particular, a construção "se e somente se" é raramente usada na linguagem comum. Em vez disso, bicondicionais são freqüentemente expressas usando a construção "se, então" ou "somente se". A outra parte do "se e somente se" fica implícita. Ou seja, a proposição oposta está implícita, mas não escrita (ou falada). Considere, por exemplo, a frase em português "Se terminar o almoço, então você pode comer a sobremesa". Essa frase tem significado exato "Você pode comer a sobremesa se, e somente se, terminar o almoço". Essa proposição é logicamente equivalente às duas proposições "Se terminar o almoço, então você pode comer a sobremesa" e "Você pode comer sobremesa somente se terminar o almoço". Como temos essa imprecisão na linguagem natural, precisamos assumir que uma proposição condicional na linguagem natural inclui sua oposta. Como a precisão é essencial em matemática e em lógica, vamos sempre fazer distinção entre condicional e bicondicional.

Tabelas-Verdade para Proposições Compostas

Demo

Agora, já estão introduzidos os quatro importantes conectivos lógicos — conjunções, disjunções, condicional e bicondicional —, assim como as negações. Podemos usar esses conectivos para construir proposições mais complicadas que envolvem qualquer número de variáveis proposicionais. Você pode usar tabelas-verdade para determinar o valor-verdade dessas proposições, como ilustrado no Exemplo 11. Usamos uma coluna para achar o valor-verdade de cada expressão composta que ocorre na proposição composta original, exatamente como está construída. O valor-verdade da proposição composta para cada combinação de valores-verdade das variáveis proposicionais é expresso na última coluna da tabela.

EXEMPLO 11 Construa a tabela-verdade da proposição composta

$$(p \vee \neg q) \rightarrow (p \wedge q).$$

Solução: Como essa proposição envolve apenas duas variáveis proposicionais p e q, existem quatro linhas nessa tabela, correspondentes às combinações dos valores-verdade VV, VF, FV e FF. As primeiras duas colunas são usadas para os valores-verdade de p e q, respectivamente. Na terceira coluna, achamos os valores-verdade de $\neg q$, necessários para encontrar os valores de $p \vee \neg q$, que podem ser achados na quarta coluna. Os valores-verdade de $p \wedge q$ são encontrados na quinta coluna. Finalmente, os valores-verdade da proposição composta $(p \vee \neg q) \rightarrow (p \wedge q)$ são encontrados na última coluna. A tabela-verdade resultante é mostrada na Tabela 7. ◀

Prioridade de Operadores Lógicos

Podemos construir proposições compostas usando a negação e os operadores lógicos já definidos antes. Vamos geralmente usar parênteses para especificar a ordem em que os operadores lógicos são aplicados em uma proposição composta. Por exemplo, $(p \vee q) \wedge (\neg r)$ é a conjunção de

TABELA 7 A Tabela-Verdade de $(p \vee \neg q) \rightarrow (p \wedge q)$.

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
V	V	F	V	V	V
V	F	V	V	F	F
F	V	F	F	F	V
F	F	V	V	F	F

TABELA 8
Prioridade do Operador Lógico.

Operador	*Prioridade*
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5

$p \vee q$ e $\neg r$. No entanto, para reduzir o número de parênteses, especificamos que a negação é aplicada antes de qualquer outro operador. Isso significa que $\neg p \wedge q$ é a conjunção de $\neg p$ e q, não a negação da conjunção de p e q.

Outra regra geral de prioridade é que a conjunção tem prioridade sobre a disjunção, então $p \wedge q \vee r$ significa $(p \wedge q) \vee r$ em vez de $p \wedge (q \vee r)$. Como essa regra pode ser mais difícil de ser lembrada, vamos continuar usando parênteses até que a ordem fique clara.

Finalmente, é uma regra aceita que a condicional e a bicondicional têm prioridade menor que a conjunção e a disjunção. Conseqüentemente, $p \vee q \rightarrow r$ é o mesmo que $(p \vee q) \rightarrow r$. Vamos usar parênteses quando a prioridade dos conectivos condicional e bicondicional estiver em ordem inversa; tal seqüência será: o condicional deve ter prioridade maior que o bicondicional. A Tabela 8 mostra os níveis de prioridade dos operadores lógicos, $\neg$, $\wedge$, $\vee$, $\rightarrow$ e $\leftrightarrow$.

Traduzindo Sentenças em Português

Existem muitas razões para traduzir sentenças em português para expressões que envolvem variáveis proposicionais e conectivos lógicos. Em particular, o português (e muitas outras linguagens humanas) é freqüentemente ambíguo. Traduzir sentenças como proposições compostas (e outros tipos de expressões lógicas, as quais vamos introduzir mais tarde neste capítulo) acaba com essa ambigüidade. Note que isso pode envolver um conjunto de fatos assumidos com base no significado de cada sentença. Mais ainda, uma vez traduzidas do português para expressões lógicas, podemos analisar seus valores-verdade, manipulá-las e ainda usar regras de inferência (as quais discutiremos na Seção 1.5) para raciocinar sobre elas.

Para ilustrar o processo de tradução de uma sentença do português para uma expressão lógica, considere os exemplos 12 e 13.

EXEMPLO 12 Como podemos traduzir esta sentença do português para expressões lógicas?

"Você pode acessar a Internet a partir deste campus somente se você é um expert em ciência da computação ou não é um novato."

Exemplos Extras

Solução: Existem muitas maneiras de traduzir essa sentença para uma expressão lógica. Por exemplo, poderíamos representá-la por uma simples variável proposicional p, mas isso pode não ser usual quando queremos analisar o significado dela ou raciocinar sobre ela. No entanto, podemos usar variáveis proposicionais para cada sentença que forma a proposição e determinar os conectivos lógicos que devem estar entre elas. Em particular, podemos usar a, c e f representando "Você pode acessar a Internet a partir deste campus", "Você é um expert em ciências da computação" e "Você é um novato", respectivamente. Note que "somente se" é uma forma de a condicional ser expressa, então podemos representar a sentença por

$$a \rightarrow (c \vee \neg f).$$ ◀

EXEMPLO 13 Como podemos traduzir esta sentença do português para expressões lógicas?

"Você pode pular de pára-quedas se você tem autorização de seus pais ou se tem mais de 18 anos."

Solução: Sejam q, r e s as representações de "Você pode pular de pára-quedas", "Você tem autorização de seus pais" e "Você tem mais de 18 anos", respectivamente. Então, a sentença pode ser traduzida por

$$(r \wedge \neg s) \rightarrow \neg q.$$

É claro que existem muitas outras maneiras de traduzir a sentença, mas essa já é suficiente. ◀

Sistemas de Especificações

A tradução de sentenças da linguagem natural para expressões lógicas é uma parte essencial para a especificação de sistemas de hardware e sistemas de software. Sistemas e engenheiros de software tomam afirmações em linguagem natural e produzem especificações precisas e sem ambigüidade que podem ser usadas como base de um sistema de desenvolvimento. O Exemplo 14 mostra como proposições compostas podem ser usadas nesse processo.

EXEMPLO 14 Expresse a especificação "A resposta automática não pode ser enviada quando o sistema está sobrecarregado", usando conectivos lógicos.

Exemplos Extras

Solução: Um meio de traduzir é tomar p como "A resposta automática pode ser enviada" e q como "O sistema está sobrecarregado". Então, $\neg p$ representa "Não é o caso de a resposta automática poder ser enviada" ou "A resposta automática não pode ser enviada". Conseqüentemente, nossa especificação pode ser representada por $q \rightarrow \neg p$. ◀

Sistemas de especificações devem ser **consistentes**, ou seja, não podem conter especificações conflitantes que possam ser usadas para derivar uma contradição. Quando as especificações não são consistentes, pode não haver um meio de desenvolver um sistema que satisfaça todas as especificações.

EXEMPLO 15 Determine se este sistema de especificações é consistente:

"A mensagem de diagnóstico é armazenada no buffer ou é retransmitida."

"A mensagem de diagnóstico não é armazenada no buffer."

"Se a mensagem de diagnóstico é armazenada no buffer, então ela é retransmitida."

Solução: Para determinar se esse sistema é consistente, primeiro vamos reescrevê-lo como expressões lógicas. Seja p "A mensagem de diagnóstico é armazenada no buffer" e q "A mensagem de diagnóstico é retransmitida". As especificações podem ser escritas como $p \vee q$, $\neg p$ e $p \rightarrow q$. Uma valoração que torna as três especificações verdadeiras deve ter p falsa para que $\neg p$ seja verdadeira. Como queremos que $p \vee q$ seja verdadeira e temos p falsa, devemos ter q verdadeira. Ainda como $p \rightarrow q$ é verdadeira quando p é falsa e q é verdadeira, concluímos que essas especificações são consistentes porque são verdadeiras quando p é falsa e q é verdadeira. Poderíamos chegar à mesma conclusão analisando as tabelas-verdade e examinando as quatro possibilidades de valores-verdade para p e q. ◀

EXEMPLO 16 O sistema de especificações do Exemplo 15 continua consistente se adicionarmos a especificação "A mensagem de diagnóstico não é retransmitida"?

Solução: Verificando o Exemplo 15, notamos que o sistema é consistente se p é falsa e q é verdadeira, mas a nova especificação representa $\neg q$, que somente é verdadeira se q é falsa. Conseqüentemente, esse novo sistema é inconsistente. ◀

Buscadores Booleanos

Links

Conectivos lógicos são largamente usados em buscadores de grandes conjuntos de informações, tais como índices de páginas da Internet. Como esses buscadores utilizam técnicas da lógica proposicional, eles são chamados de **buscadores booleanos**.

Em buscadores booleanos, o conectivo *E (AND)* é usado para encontrar informações que contenham ambos os termos procurados; o conectivo *OU (OR)* é usado para encontrar informações que contenham um ou ambos os termos procurados; e o conectivo *NÃO (NOT)*, às vezes escrito *E NÃO (AND NOT)*, é usado para excluir alguma informação que contenha esse termo na procura. Um estudo cuidadoso de como são usados os conectivos lógicos é necessário quando buscadores booleanos são usados para localizar alguma informação de interesse potencial. O Exemplo 17 ilustra como funcionam os buscadores booleanos.

EXEMPLO 17 **Pesquisando Páginas da Internet** A maioria dos buscadores na Web, os quais usualmente podem nos ajudar a encontrar páginas da Internet sobre algum objeto específico, utiliza técnicas de buscadores booleanos. Por exemplo, usando um buscador booleano para achar uma página da Web sobre universidades em São Paulo, devemos procurar por páginas que trabalhem com SÃO *AND* PAULO *AND* UNIVERSIDADES. Os resultados dessa busca incluirão as páginas que contêm as três palavras SÃO, PAULO e UNIVERSIDADES. Isso deve incluir todas as páginas de interesse, assim como páginas sobre universidades que têm algum texto sobre São Paulo. (Note que, no Google e em muitos outros buscadores, a palavra "AND" não é necessária; essa fica subentendida porque todos os termos são incluídos.) Segundo, para encontrar páginas de universidades em São Paulo ou Paraná, devemos procurar por páginas que trabalhem com (SÃO *AND* PAULO *OR* PARANÁ) *AND* UNIVERSIDADES. (*Nota*: Aqui o operador *AND* tem prioridade maior que o operador *OR*. Além disso, no Google, os termos usados devem ser SÃO PAULO *OR* PARANÁ.) O resultado dessa busca deve incluir todas as páginas com a palavra UNIVERSIDADES e/ou uma ou ambas as palavras SÃO e PAULO ou PARANÁ. Novamente, páginas com textos que incluem essas palavras, mas que não são de interesse, serão listadas. Finalmente, para encontrar páginas sobre universidades em São Paulo, não públicas, podemos primeiro fazer uma busca com SÃO *AND* PAULO *AND* UNIVERSIDADES, mas essa busca incluirá as páginas sobre as universidades também públicas; então, podemos buscar por (SÃO *AND* PAULO *AND* UNIVERSIDADES) *NOT* PÚBLICAS. O resultado listará as páginas que contêm as palavras SÃO e PAULO e UNIVERSIDADES e não contêm a palavra PÚBLICAS. (No Google e em muitos outros buscadores, a palavra "NOT" é substituída pelo sinal de subtração "–"; nesse caso, a busca seria SÃO PAULO UNIVERSIDADES – PÚBLICAS.) ◀

Exemplos Extras

Quebra-Cabeças Lógicos

Links

Enigmas que podem ser resolvidos por raciocínio lógico são chamados de **quebra-cabeças lógicos**. Resolver quebra-cabeças lógicos é um excelente meio de treinar as regras da lógica. Também os programas de computadores que devem trabalhar com raciocínio lógico freqüentemente usam conhecidos quebra-cabeças para demonstrar sua capacidade. Muitas pessoas apreciam resolver esses quebra-cabeças lógicos, os quais são publicados em livros e periódicos como atividade de recreação.

Vamos discutir dois quebra-cabeças lógicos. Começaremos com um originalmente proposto por Raymond Smullyan, um mestre desses jogos, que publicou muitos livros com quebra-cabeças desafiantes que envolvem raciocínio lógico.

EXEMPLO 18 Em [Sm78] Smullyan propôs muitos quebra-cabeças sobre uma ilha que contém dois tipos de habitantes: cavaleiros, que sempre falam a verdade, e bandidos, que sempre mentem. Você encontra duas pessoas *A* e *B*. Quem são *A* e *B*, se *A* diz "*B* é um cavaleiro" e *B* diz "Nós dois somos tipos opostos de habitantes"?

Exemplos Extras

Solução: Sejam p e q as proposições "A é um cavaleiro" e "B é um cavaleiro", respectivamente, então $\neg p$ e $\neg q$ são as afirmações que dizem ser A e B bandidos, respectivamente.

Primeiro, vamos considerar a possibilidade de A ser cavaleiro; ou seja, a proposição p é verdadeira. Se isso ocorre, então ele está falando a verdade, o que implica que B é um cavaleiro. Sendo assim, o que B fala é verdade também; no entanto, ele diz que os dois são de tipos diferentes e estamos concluindo que ambos são cavaleiros, o que é absurdo. Então, devemos pensar que A é um bandido, logo está falando mentira, e portanto B também é bandido. Este fato é plausível, pois se ele está mentindo também, então ambos devem ser habitantes do mesmo tipo, o que é verdade. Podemos, então, concluir que ambos são bandidos. ◀

Proporemos mais desses quebra-cabeças de Smullyan sobre cavaleiros e bandidos nos exercícios 55 a 59 do fim desta seção. Agora, proporemos o conhecido como **o quebra-cabeça das crianças enlameadas**, que fala de duas crianças.

EXEMPLO 19 Um pai diz aos filhos, um menino e uma menina, para brincarem no quintal sem se sujarem. No entanto, enquanto brincavam, os dois sujaram a testa de lama. Quando pararam de brincar, o pai disse: "Ao menos um de vocês está com lama na testa", e depois pediu que cada criança respondesse sim ou não à pergunta: "Você sabe se você tem lama na testa?". O pai faz essa pergunta duas vezes. O que as crianças vão responder cada vez que a pergunta for feita, assumindo que cada criança pode ver a testa da outra, mas não pode ver sua própria testa? Assuma que cada criança é honesta e que as crianças respondem à pergunta simultaneamente.

Solução: Seja s a proposição que diz que o filho tem a testa suja e d a proposição que diz que a filha tem a testa suja. Quando o pai diz que ao menos uma das duas crianças tem a testa suja, ele está afirmando que a disjunção $s \vee d$ é verdadeira. Ambas as crianças vão responder "não" na primeira vez que a pergunta for feita, pois elas estão vendo o rosto uma da outra e que está sujo; logo, acreditam que a outra está suja e não elas próprias. Ou seja, o filho diz que d é verdadeira e s é falsa. E a filha diz exatamente o contrário, d é falsa e s é verdadeira.

Depois da resposta negativa do menino, a menina conclui que sua testa está suja, já que o menino afirmou que d é verdadeira. Isso pode ser concluído, pois o pai afirmou e o menino não pode ver sua própria testa.

Links

RAYMOND SMULLYAN (NASCIDO EM 1919) Raymond Smullyan abandonou os estudos no colegial. Ele queria estudar aquilo que realmente lhe interessava e não o conteúdo padrão do colegial. Depois de passar de uma universidade para outra, ele conquistou um diploma de graduação em matemática pela Universidade de Chicago, em 1955. Ele pagou suas despesas universitárias trabalhando com mágica em festas e clubes. Ele obteve seu Ph.D. em lógica em 1959, em Princeton, orientado por Alonzo Church. Depois da graduação em Princeton, ele ensinou matemática e lógica nas Universidades de Dartmouth, de Princeton, de Yeshiva e na Universidade da Cidade de Nova York. Ele se juntou ao departamento de filosofia da Universidade de Indiana em 1981, onde é agora professor emérito.

Smullyan escreveu muitos livros sobre lógica recreacional e matemática, incluindo *Satan, Cantor, and Infinity; What Is the Name of This Book?; The Lady or the Tiger?; Alice in Puzzleland; To Mock a Mockingbird; Forever Undecided;* e *The Riddle of Scheherazade: Amazing Logic Puzzles, Ancient and Modern.* Por seus quebra-cabeças lógicos serem desafiantes e provocantes, ele é considerado um Lewis Carroll dos dias atuais. Smullyan também escreveu diversos livros sobre a lógica dedutiva aplicada ao xadrez, três coleções de ensaios filosóficos e aforismos e muitos livros avançados de lógica matemática e teoria dos conjuntos. Ele se interessa particularmente por auto-referência e tem trabalhado em alguns resultados de Gödel que mostram que é impossível escrever um programa de computador que solucione problemas matemáticos. Ele também se interessa por explicar as idéias da matemática lógica ao público em geral.

Smullyan é um músico talentoso e geralmente toca piano com sua esposa, que é uma pianista de concerto. Fazer telescópios é um de seus hobbies. Ele também se interessa por ótica e fotografia em estéreo. Seu lema: "Eu nunca tive um conflito entre o ensino e a pesquisa, assim como algumas pessoas, porque, quando estou ensinando, estou pesquisando".

TABELA 9 Tabela para os Operadores Binários *OR*, *AND* e *XOR*.

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Note que a menina pode concluir que *d* é verdadeira, pois, se *d* fosse falsa, o menino deveria concluir *s* e responder "sim", ou seja, tinha lama na própria testa. Sendo assim, ela conclui que tem lama na própria testa e responde "sim" na segunda vez que a pergunta foi feita. Com um raciocínio análogo, o menino conclui que também tem lama na testa e também responde "sim" na segunda vez que a pergunta foi feita. ◀

Lógicas e Operações Bit

Valor-Verdade	*Bit*
V	1
F	0

Computadores representam informações usando bits. Um **bit** é um símbolo com dois valores possíveis, 0 (zero) e 1 (um). O significado da palavra bit vem de *binary digit* (dígito binário), porque zeros e uns são os únicos dígitos usados na numeração binária. O conhecido estatístico John Tukey introduziu este termo em 1946. Um bit pode ser usado para representar um valor-verdade, pois existem dois valores-verdade, *verdadeiro* e *falso*. Como é costumeiramente feito, vamos usar um bit 1 para representar o verdadeiro e um bit 0 para representar o falso. Ou seja, 1 representa V, 0 representa F. Uma variável é chamada de **variável booleana** se seu valor puder ser verdadeiro ou falso. Conseqüentemente, uma variável booleana pode ser representada por um bit.

Uma computação chamada de **operação bit** (ou **operação binária**) corresponde aos conectivos lógicos, trocando verdadeiro por 1 e falso por 0 nas tabelas-verdade dos operadores $\wedge$, $\vee$ e $\oplus$; a Tabela 9 mostra as operações binárias obtidas. Também vamos usar a notação *AND*, *OR* e *XOR* para os operadores $\wedge$, $\vee$ e $\oplus$, como em algumas linguagens computacionais.

Informações são freqüentemente representadas usando seqüências binárias (bit strings), que são seqüências de zeros e uns. Quando isso é feito, operações nas seqüências binárias podem ser usadas para manipular essas informações.

DEFINIÇÃO 7 Uma *seqüência binária* é uma seqüência de zero ou mais bits. A *extensão* dessa seqüência é o número de dígitos (bits) que ela contém.

EXEMPLO 20 101010011 é uma seqüência binária de comprimento nove. ◀

Podemos estender operações bit para seqüências binárias. Definimos a **seqüência binária tipo *OU***, a **seqüência binária tipo *E*** e a **seqüência binária tipo *OU*-exclusivo** (**bitwise *OR*, bitwise *AND* e bitwise *XOR***) de duas seqüências binárias de mesmo comprimento como aquela que tem como seus bits os bits correspondentes ao *OU*, *E* e *OU*-exclusivo para os respectivos dígitos das duas seqüências originais. Usaremos os símbolos dos operadores $\vee$, $\wedge$ e $\oplus$ para representar as seqüências binárias tipo *OU*, tipo *E* e tipo *OU*-exclusivo, respectivamente. Vamos ilustrar essas operações com o Exemplo 21.

EXEMPLO 21 Encontre a seqüência binária tipo *OU*, a seqüência binária tipo *E* e a seqüência binária tipo *OU*-exclusivo das seqüências 01 1011 0110 e 11 0001 1101. (Aqui, e em todo o livro, as seqüências serão separadas em blocos de quatro bits para facilitar a leitura.)

Solução: As seqüências são:

01 1011 0110	
11 0001 1101	
11 1011 1111	*OU*
01 0001 0100	*E*
10 1010 1011	*OU*-exclusivo

◀

Exercícios

1. Quais dessas sentenças são proposições? Quais são os valores-verdade das que são proposições?

a) Curitiba é a capital do Paraná.

b) Joinville é a capital de Santa Catarina.

c) $2 + 3 = 5$.

d) $5 + 7 = 10$.

e) $x + 2 = 11$.

f) Responda esta questão.

2. Quais destas são proposições? Quais são os valores-verdade das que são proposições?

a) Não ultrapasse.

b) Que horas são?

c) Não há moscas pretas em Brasília.

d) $4 + x = 5$.

e) A lua é feita de queijo verde.

f) $2^n \geq 100$.

3. Qual é a negação de cada proposição a seguir?

a) Hoje é quinta-feira.

b) Não há poluição em São Paulo.

c) $2 + 1 = 3$.

d) O verão no Rio é quente e ensolarado.

4. Considere que p e q são proposições:

p: Eu comprei um bilhete de loteria esta semana.

q: Eu ganhei a bolada de um milhão de dólares na sexta-feira.

Expresse cada uma dessas proposições em uma sentença em português.

a) $\neg p$ **b)** $p \vee q$ **c)** $p \rightarrow q$

d) $p \wedge q$ **e)** $p \leftrightarrow q$ **f)** $\neg p \rightarrow \neg q$

g) $\neg p \wedge \neg q$ **h)** $\neg p \vee (p \wedge q)$

5. Considere que p e q são as proposições: "Nadar na praia em Nova Jersey é permitido." e "Foram descobertos tubarões perto da praia.", respectivamente. Expresse cada uma dessas proposições compostas como uma sentença em português.

a) $\neg q$ **b)** $p \wedge q$ **c)** $\neg p \vee q$

d) $p \rightarrow \neg q$ **e)** $\neg q \rightarrow p$ **f)** $\neg p \rightarrow \neg q$

g) $p \leftrightarrow \neg q$ **h)** $\neg p \wedge (p \vee \neg q)$

Links

JOHN WILDER TUKEY (1915–2000) Tukey, nascido em New Bedford, Massachusetts, era filho único. Seus pais, ambos professores, decidiram que a educação em casa seria a melhor opção para o desenvolvimento de seu potencial. Sua educação formal iniciou-se na Universidade de Brown, onde estudou matemática e química. Tukey recebeu o mestrado em química da Brown e continuou seus estudos em Princeton. Com o início da Segunda Guerra Mundial, ele se juntou ao Fire Control Research Office, onde começou a trabalhar com estatística. Tukey, em suas pesquisas com estatísticas, impressionou muitos estatísticos com suas habilidades. Em 1945, com o fim da guerra, Tukey retornou ao departamento de matemática de Princeton como professor de estatística e também associou-se ao laboratório AT&T. Tukey fundou o Departamento de Estatística da Princeton em 1966 e foi seu primeiro catedrático. Fez contribuições significativas em muitas áreas da estatística, incluindo análise de variantes, estimativa do espectro das séries de tempo, inferências sobre valor de um grupo de parâmetros de um experimento e filosofia da estatística. Entretanto, ele é muito conhecido por sua invenção, em parceria com J. W. Cooley, da transformação rápida de Fourier. Além de suas contribuições em estatística, Tukey é visto como um habilidoso conhecedor de Wordsmith; tem o crédito de cunhar os termos *bit* e *software*.

Tukey contribuiu com sua visão e conhecimento servindo o Comitê Consultivo de Ciência do Presidente. Ele liderou diversos comitês importantes, lidando com meio ambiente, educação, química e saúde. Ele também serviu nos comitês de desarmamento nuclear. Tukey recebeu muitos prêmios, incluindo a Medalha Nacional de Ciência.

NOTA HISTÓRICA Há muitas outras denominações para dígito binário, como *binit* e *bigit*, mas nunca foram mundialmente aceitas. A adoção da palavra bit é ligada a sua semelhança com a palavra em inglês. Para uma descrição da cunhagem da palavra *bit* por Tukey, veja a revista *Anais da História da Computação*, de abril de 1984.

6. Considere que p e q são proposições: "A eleição está decidida" e "Os votos foram contados", respectivamente. Expresse cada uma destas proposições compostas como uma sentença em português.

a) $\neg p$ **b)** $p \vee q$
c) $\neg p \wedge q$ **d)** $q \rightarrow p$
e) $\neg q \rightarrow \neg p$ **f)** $\neg p \rightarrow \neg q$
g) $p \leftrightarrow q$ **h)** $\neg q \vee (\neg p \wedge q)$

7. Considere que p e q são proposições:

p: Está abaixo de zero.
q: Está nevando.

Escreva estas proposições usando p, q e conectivos lógicos.

a) Está abaixo de zero e nevando.
b) Está abaixo de zero, mas não está nevando.
c) Não está abaixo de zero e não está nevando.
d) Está ou nevando ou abaixo de zero (ou os dois).
e) Se está abaixo de zero, está também nevando.
f) Está ou nevando ou abaixo de zero, mas não está nevando se estiver abaixo de zero.
g) Para que esteja abaixo de zero é necessário, e suficiente, que esteja nevando.

8. Considere que p, q e r são as proposições:

p : Você está com gripe.
q : Você perde a prova final.
r : Você foi aprovado no curso.

Expresse cada uma destas proposições compostas como uma sentença em português.

a) $p \rightarrow q$ **b)** $\neg q \leftrightarrow r$
c) $q \rightarrow \neg r$ **d)** $p \vee q \vee r$
e) $(p \rightarrow \neg r) \vee (q \rightarrow \neg r)$
f) $(p \wedge q) \vee (\neg q \wedge r)$

9. Considere que p e q são proposições:

p : Você dirige a mais de 104 km/h.
q : Você recebe uma multa por excesso de velocidade.

Escreva estas proposições usando p, q e conectivos lógicos.

a) Você não dirige a mais de 104 km/h.
b) Você dirige a mais de 104 km/h, mas não recebe uma multa por excesso de velocidade.
c) Você receberá uma multa por excesso de velocidade, se você dirigir a mais de 104 km/h.
d) Se você não dirigir a mais de 104 km/h, você não receberá uma multa por excesso de velocidade.
e) Dirigir a mais de 104 km/h é suficiente para receber uma multa por excesso de velocidade.
f) Você recebe uma multa por excesso de velocidade, mas você não dirige a mais de 104 km/h.
g) Sempre que receber uma multa por excesso de velocidade, você estará dirigindo a mais de 104 km/h.

10. Considere que p, q e r são proposições:

p : Você tira um A no exame final.
q : Você faz todos os exercícios deste livro.
r : Você tira um A nesta matéria.

Escreva estas proposições usando p, q, r e conectivos lógicos.

a) Você tira um A nesta matéria, mas não faz todos os exercícios deste livro.
b) Você tira um A no exame final, faz todos os exercícios deste livro e tira um A nesta matéria.
c) Tirar um A nesta matéria é necessário para tirar um A no exame final.
d) Você tira um A no exame final, mas não faz todos os exercícios deste livro; no entanto, tira um A nesta matéria.
e) Tirar um A no exame final e fazer todos os exercícios deste livro é suficiente para tirar A nesta matéria.
f) Você vai tirar um A nesta matéria se e somente se você fizer todos os exercícios deste livro ou você tirar um A no exame final.

11. Considere que p, q e r são proposições:

p : Ursos-cinzentos são vistos na área.
q : Fazer caminhada na trilha é seguro.
r : As bagas estão maduras ao longo da trilha.

Escreva estas proposições usando p, q, r e conectivos lógicos.

a) As bagas estão maduras ao longo da trilha, mas os ursos-cinzentos não são vistos na área.
b) Ursos-cinzentos não são vistos na área e fazer caminhada na trilha é seguro, mas as bagas estão maduras ao longo da trilha.
c) Se as bagas estão maduras ao longo da trilha, fazer caminhada é seguro se e somente se os ursos-cinzentos não forem vistos na área.
d) Não é seguro fazer caminhada na trilha, mas os ursos-cinzentos não são vistos na área e as bagas ao longo da trilha estão maduras.
e) Para a caminhada ser segura, é necessário, mas não suficiente, que as bagas não estejam maduras ao longo da trilha e que os ursos-cinzentos não sejam vistos na área.
f) Caminhada não é segura ao longo da trilha sempre que os ursos-cinzentos são vistos na área e as bagas estão maduras ao longo da trilha.

12. Determine se estes bicondicionais são verdadeiros ou falsos.

a) $2 + 2 = 4$ se e somente se $1 + 1 = 2$.
b) $1 + 1 = 2$ se e somente se $2 + 3 = 4$.
c) $1 + 1 = 3$ se e somente se macacos puderem voar.
d) $0 > 1$ se e somente se $2 > 1$.

13. Determine se cada uma destas proposições condicionais é verdadeira ou falsa.

a) Se $1 + 1 = 2$, então $2 + 2 = 5$.
b) Se $1 + 1 = 3$, então $2 + 2 = 4$.
c) Se $1 + 1 = 3$, então $2 + 2 = 5$.
d) Se macacos puderem voar, então $1 + 1 = 3$.

14. Determine se cada uma destas proposições condicionais é verdadeira ou falsa.

a) Se $1 + 1 = 3$, então unicórnios existem.
b) Se $1 + 1 = 3$, então cachorros podem voar.
c) Se $1 + 1 = 2$, então cachorros podem voar.
d) Se $2 + 2 = 4$, então $1 + 2 = 3$.

15. Para cada uma destas sentenças, determine se o ou é inclusivo ou exclusivo. Explique sua resposta.
- **a)** Café ou chá vem com o jantar.
- **b)** Uma senha deve ter ao menos três dígitos ou oito caracteres de comprimento.
- **c)** O pré-requisito para o curso é um curso em teoria dos números ou um curso em criptografia.
- **d)** Você pode jogar usando dólares americanos ou euros.

16. Para cada uma destas sentenças, determine se o ou é inclusivo ou exclusivo. Explique sua resposta.
- **a)** Experiência em C++ ou Java é necessária.
- **b)** O almoço inclui sopa ou salada.
- **c)** Para entrar no país, é necessário um passaporte ou um cartão de registro eleitoral.
- **d)** Publique ou sucumba.

17. Para cada sentença, identifique o que significa a sentença, se o ou é inclusivo (ou seja, uma disjunção) ou exclusivo. Quais dos significados do ou você pensa ser intencional?
- **a)** Para cursar matemática discreta, você deve ter tido cálculo ou um curso de ciência da computação.
- **b)** Quando você compra um novo carro da Companhia Acme Motor, você pega de volta $ 2.000 ou um empréstimo de 2%.
- **c)** Jantar para dois inclui dois itens da coluna A ou três itens da coluna B.
- **d)** A escola fecha se cair mais de dois pés de neve ou se a sensação térmica estiver abaixo de −100.

18. Escreva cada uma destas proposições na forma "se p, então q" em português. (*Dica*: Recorra à lista de maneiras comuns de expressar proposições condicionais inserida nesta seção.)
- **a)** É necessário lavar o carro do chefe para ser promovido.
- **b)** Ventos do sul implicam um degelo primaveril.
- **c)** Uma condição suficiente para a garantia ser válida é que você tenha comprado o computador em menos de um ano.
- **d)** Leo é pego sempre que ele trapaceia.
- **e)** Você pode acessar o site apenas se você pagar uma taxa de assinatura.
- **f)** Escolha as companhias certas, conhecendo as pessoas certas.
- **g)** Carol fica enjoada sempre que está em um barco.

19. Escreva cada uma destas proposições na forma "se p, então q" em português. (*Dica*: Recorra à lista de maneiras comuns de expressar proposições condicionais inserida nesta seção.)
- **a)** Neva sempre que o vento sopra do nordeste.
- **b)** As macieiras florescerão se continuar quente por uma semana.
- **c)** O Palmeiras ganhar o campeonato implica derrotar o São Paulo.
- **d)** É necessário andar 8 milhas para chegar ao topo do "Pico Long".
- **e)** Para conseguir mandato como professor, é suficiente ser famoso mundialmente.
- **f)** Se você dirigir por mais de 400 milhas, terá de comprar gasolina.
- **g)** Sua garantia é válida apenas se você comprou seu aparelho de som em menos de 90 dias.
- **h)** Jan nadará a menos que a água esteja muito fria.

20. Escreva cada uma destas proposições na forma "se p, então q" em português. (*Dica*: Recorra à lista de maneiras comuns de expressar proposições condicionais inserida nesta seção.)
- **a)** Eu lembrarei de enviar para você o endereço apenas se você me mandar um e-mail.
- **b)** Para ser um cidadão americano, é suficiente que você tenha nascido nos Estados Unidos.
- **c)** Se você mantiver seu livro teórico, ele será uma referência útil em seus cursos futuros.
- **d)** O São Paulo vencerá o Campeonato Brasileiro se seu goleiro jogar bem.
- **e)** Conseguir o emprego implica você ter as melhores credenciais.
- **f)** Haverá erosão na praia sempre que houver uma tempestade.
- **g)** Para ter uma senha válida, é necessário que inicie uma conexão no servidor.
- **h)** Você alcançará o cume a menos que você comece a escalada muito tarde.

21. Escreva cada uma destas proposições na forma "p se e somente se q" em português.
- **a)** Se está calor lá fora, você compra um sorvete e se você compra um sorvete é porque está calor lá fora.
- **b)** Para que você ganhe na loteria, é necessário e suficiente que você tenha o único bilhete premiado.
- **c)** Você será promovido apenas se você tiver contatos, e você só terá contatos se for promovido.
- **d)** Se você assistir à televisão sua mente se deteriorará, e vice-versa.
- **e)** Os trens atrasam exatamente nos dias em que eu os pego.

22. Escreva cada uma destas proposições na forma "p se e somente se q" em português.
- **a)** Para que você obtenha um A neste curso, é necessário e suficiente que você aprenda como resolver problemas de matemática discreta.
- **b)** Se você ler jornal todos os dias, você estará informado, e vice-versa.
- **c)** Chove se é final de semana, e é final de semana quando chove.
- **d)** Você poderá ver o feiticeiro apenas se ele não estiver escondido, e o feiticeiro não estará escondido apenas se você puder vê-lo.

23. Determine a oposta, a contrapositiva e a inversa de cada uma das proposições condicionais.
- **a)** Se nevar hoje, esquiarei amanhã.
- **b)** Eu venho à aula sempre que há uma prova.
- **c)** Um inteiro positivo é um primo apenas se não tem divisores além de 1 e dele mesmo.

24. Determine a oposta, a contrapositiva e a inversa de cada uma das proposições condicionais.
- **a)** Se nevar esta noite, então ficarei em casa.
- **b)** Eu vou à praia sempre que faz um dia ensolarado de verão.
- **c)** Quando me deito tarde, é necessário que eu durma até o meio-dia.

25. Quantas linhas aparecem em uma tabela-verdade para cada uma destas proposições compostas?
- **a)** $p \rightarrow \neg p$

b) $(p \vee \neg r) \wedge (q \vee \neg s)$
c) $q \vee p \vee \neg s \vee \neg r \vee \neg t \vee u$
d) $(p \wedge r \wedge t) \leftrightarrow (q \wedge t)$

26. Quantas linhas aparecem em uma tabela-verdade para cada uma destas proposições compostas?
a) $(q \rightarrow \neg p) \vee (\neg p \rightarrow \neg q)$
b) $(p \vee \neg t) \wedge (p \vee \neg s)$
c) $(p \rightarrow r) \vee (\neg s \rightarrow \neg t) \vee (\neg u \rightarrow v)$
d) $(p \wedge r \wedge s) \vee (q \wedge t) \vee (r \wedge \neg t)$

27. Construa uma tabela-verdade para cada uma destas proposições compostas.
a) $p \wedge \neg p$ **b)** $p \vee \neg p$
c) $(p \vee \neg q) \rightarrow q$ **d)** $(p \vee q) \rightarrow (p \wedge q)$
e) $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$
f) $(p \rightarrow q) \rightarrow (q \rightarrow p)$

28. Construa uma tabela-verdade para cada uma destas proposições compostas.
a) $p \rightarrow \neg p$ **b)** $p \leftrightarrow \neg p$
c) $p \oplus (p \vee q)$ **d)** $(p \wedge q) \rightarrow (p \vee q)$
e) $(q \rightarrow \neg p) \leftrightarrow (p \leftrightarrow q)$
f) $(p \leftrightarrow q) \oplus (p \leftrightarrow \neg q)$

29. Construa uma tabela-verdade para cada uma destas proposições compostas.
a) $(p \vee q) \rightarrow (p \oplus q)$ **b)** $(p \oplus q) \rightarrow (p \wedge q)$
c) $(p \vee q) \oplus (p \wedge q)$ **d)** $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow q)$
e) $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow \neg r)$
f) $(p \oplus q) \rightarrow (p \oplus \neg q)$

30. Construa uma tabela-verdade para cada uma destas proposições compostas.
a) $p \oplus p$ **b)** $p \oplus \neg p$
c) $p \oplus \neg q$ **d)** $\neg p \oplus \neg q$
e) $(p \oplus q) \vee (p \oplus \neg q)$ **f)** $(p \oplus q) \wedge (p \oplus \neg q)$

31. Construa uma tabela-verdade para cada uma destas proposições compostas.
a) $p \rightarrow \neg q$ **b)** $\neg p \leftrightarrow q$
c) $(p \rightarrow q) \vee (\neg p \rightarrow q)$ **d)** $(p \rightarrow q) \wedge (\neg p \rightarrow q)$
e) $(p \leftrightarrow q) \vee (\neg p \leftrightarrow q)$
f) $(\neg p \leftrightarrow \neg q) \leftrightarrow (p \leftrightarrow q)$

32. Construa uma tabela-verdade para cada uma destas proposições compostas.
a) $(p \vee q) \vee r$ **b)** $(p \vee q) \wedge r$
c) $(p \wedge q) \vee r$ **d)** $(p \wedge q) \wedge r$
e) $(p \vee q) \wedge \neg r$ **f)** $(p \wedge q) \vee \neg r$

33. Construa uma tabela-verdade para cada uma destas proposições compostas.
a) $p \rightarrow (\neg q \vee r)$
b) $\neg p \rightarrow (q \rightarrow r)$
c) $(p \rightarrow q) \vee (\neg p \rightarrow r)$
d) $(p \rightarrow q) \wedge (\neg p \rightarrow r)$
e) $(p \leftrightarrow q) \vee (\neg q \leftrightarrow r)$
f) $(\neg p \leftrightarrow \neg q) \leftrightarrow (q \leftrightarrow r)$

34. Construa uma tabela-verdade para $((p \rightarrow q) \rightarrow r) \rightarrow s$.

35. Construa uma tabela-verdade para $(p \leftrightarrow q) \leftrightarrow (r \leftrightarrow s)$.

36. Qual o valor de x depois que cada uma destas proposições se depararem com um programa de computador, se $x = 1$ antes de a proposição ser alcançada?
a) **if** $1 + 2 = 3$ **then** $x := x + 1$
b) **if** $(1 + 1 = 3)$ *OR* $(2 + 2 = 3)$ **then** $x := x + 1$
c) **if** $(2 + 3 = 5)$ *AND* $(3 + 4 = 7)$ **then** $x := x + 1$
d) **if** $(1 + 1 = 2)$ *XOR* $(1 + 2 = 3)$ **then** $x := x + 1$
e) **if** $x < 2$ **then** $x := x + 1$

37. Encontre a disjunção binária *OR*, a conjunção binária *AND* e a disjunção binária exclusiva *XOR* de cada um destes pares de seqüências de bit.
a) 101 1110, 010 0001
b) 1111 0000, 1010 1010
c) 00 0111 0001, 10 0100 1000
d) 11 1111 1111, 00 0000 0000

38. Dê os valores de cada uma destas expressões.
a) $1\ 1000 \wedge (0\ 1011 \vee 1\ 1011)$
b) $(0\ 1111 \wedge 1\ 0101) \vee 0\ 1000$
c) $(0\ 1010 \oplus 1\ 1011) \oplus 0\ 1000$
d) $(1\ 1011 \vee 0\ 1010) \wedge (1\ 0001 \vee 1\ 1011)$

Lógicas Fuzzy são utilizadas em inteligência artificial. Na lógica fuzzy, a proposição tem um valor-verdade que é um número entre 0 e 1, inclusive. Uma proposição com valor-verdade 0 é falsa e uma com valor-verdade 1 é verdadeira. Valores entre 0 e 1 indicam variantes de grau de verdade. Por exemplo, o valor-verdade 0,8 pode ser indicado para uma proposição "Fred é feliz", porque ele é feliz na maior parte do tempo; e o valor-verdade 0,4 pode ser indicado para a proposição "John é feliz", porque ele é feliz menos que a metade do tempo.

39. O valor-verdade da negação de uma proposição em lógica fuzzy é 1 menos o valor-verdade da proposição. Quais são os valores-verdade das proposições "Fred não é feliz" e "John não é feliz"?

40. O valor-verdade da conjunção de duas proposições em lógica fuzzy é o mínimo dos valores-verdade de duas proposições. Quais são os valores verdade das proposições "Fred e John são felizes" e "Nem Fred nem John são felizes"?

41. O valor-verdade da disjunção de duas proposições em lógica fuzzy é o máximo dos valores-verdade de duas proposições. Quais são os valores-verdade das proposições "Fred é feliz ou John é feliz" e "Fred não é feliz ou John não é feliz"?

***42.** A asserção "Esta declaração é falsa" é uma proposição?

***43.** A *n*ésima proposição em uma lista de 100 proposições é "Exatamente *n* dessas proposições nesta lista são falsas".
a) Quais as conclusões que você pode obter dessas proposições?
b) Responda ao item (a) se a *n*ésima proposição for "No mínimo *n* dessas proposições nesta lista são falsas".
c) Responda ao item (b), assumindo que a lista contém 99 proposições.

44. Uma antiga lenda siciliana diz que o barbeiro em uma cidade longínqua, que pode ser alcançada apenas se for percorrida uma perigosa estrada na montanha, barbeia aquelas pessoas e apenas aquelas que não podem se barbear sozinhas. Pode haver lá esse barbeiro?

45. Cada habitante de uma vila longínqua sempre diz a verdade ou sempre mente. Um habitante dela dará apenas como resposta um "Sim" ou um "Não" para a pergunta que um turista fizer. Suponha que você seja um turista que visita essa área e que chegue a uma bifurcação na estrada. Um lado leva até as ruínas que você quer visitar; o outro, às profundezas de uma floresta. Um habitante dessa vila está parado nessa bifurcação. Que pergunta você pode fazer ao habitante para determinar qual lado pegar?

46. Um explorador foi capturado por um grupo de canibais. Há dois tipos de canibais: aqueles que sempre dizem a verdade e aqueles que sempre mentem. Os canibais farão um churrasco com o explorador a menos que ele possa determinar se um canibal em particular sempre mente ou sempre diz a verdade. O canibal permite que ele faça apenas uma pergunta.

a) Explique por que a pergunta "Você é um mentiroso" não é válida.

b) Descubra a pergunta que o explorador pode fazer para determinar se o canibal sempre mente ou sempre diz a verdade.

47. Expresse estas especificações de sistema usando as proposições p "A mensagem é verificada contra vírus" e q "A mensagem é enviada de um sistema desconhecido", juntamente com conectivos lógicos.

a) "A mensagem é verificada contra vírus sempre que a mensagem é enviada de um sistema desconhecido."

b) "A mensagem foi enviada de um sistema desconhecido, mas não foi verificada contra vírus."

c) "É necessário verificar a mensagem contra vírus sempre que ela for enviada de um sistema desconhecido."

d) "Quando a mensagem não é enviada de um sistema desconhecido, não é verificada contra vírus."

48. Expresse este sistema de especificações usando as proposições p "O usuário entra com uma senha válida", q "O acesso é liberado" e r "O usuário pagou a taxa de assinatura", juntamente com conectivos lógicos.

a) "O usuário pagou a taxa de assinatura, mas não entra com uma senha válida."

b) "O acesso é liberado sempre que o usuário pagar a taxa de assinatura e entrar com uma senha válida."

c) "O acesso é negado se o usuário não pagou a taxa de assinatura."

d) "Se o usuário não entrar com uma senha válida, mas pagar a taxa de assinatura, então o acesso é liberado."

49. Este sistema de especificações é consistente? "O sistema está em um estado de multiuso se e somente se estiver operando normalmente. Se o sistema está operando normalmente, o kernel está funcionando. O kernel não está funcionando ou o sistema está no modo de interrupção. Se o sistema não está em um estado de multiuso, então está em um modo de interrupção. O sistema não está no modo de interrupção."

50. Este sistema de especificações é consistente? "Sempre que o software do sistema está sendo atualizado, os usuários não podem acessar os arquivos do sistema. Se os usuários podem acessar os arquivos do sistema, então eles podem salvar novos arquivos. Se os usuários não podem salvar novos arquivos, então o software do sistema não está sendo atualizado."

51. Este sistema de especificações é consistente? "O roteador pode mandar pacotes para o sistema principal apenas se ele suportar um novo espaço de endereço. Para o roteador suportar o novo espaço de endereço, é necessário que a última liberação do software seja instalada. O roteador pode mandar pacotes ao sistema principal se a última liberação do software estiver instalada. O roteador não comporta o novo espaço."

52. Este sistema de especificações é consistente? "Se o sistema de arquivos não está bloqueado, então novas mensagens entraram em fila. Se o sistema de arquivos não está bloqueado, então o sistema está funcionando normalmente, e vice-versa. Se novas mensagens não estão entrando em fila, então serão enviadas para uma central de armazenamento de mensagens. Se o sistema de arquivos não está bloqueado, então as novas mensagens serão enviadas para a central de armazenamento. Novas mensagens não serão enviadas para a central de armazenamento."

53. Qual busca booleana você utilizaria para procurar sites sobre praias em Nova Jersey? Qual você utilizaria se quisesse encontrar sites sobre praias na ilha de Jersey (no Canal da Mancha)?

54. Qual busca booleana você utilizaria para procurar sites sobre caminhadas no oeste da Virgínia, nos Estados Unidos? Qual busca booleana você utilizaria para procurar sites sobre caminhadas na Virgínia, mas não no oeste da Virgínia?

Os exercícios 55 a 59 são relativos aos habitantes da ilha de cavaleiros e bandidos, criada por Smullyan, onde os cavaleiros sempre dizem a verdade e os bandidos sempre mentem. Você encontra duas pessoas, A e B. Determine, se possível, quem são A e B se eles conduzirem você nos caminhos descritos. Se não puder determinar quem são essas duas pessoas, você pode tirar alguma conclusão?

55. A diz: "Ao menos um de nós é um bandido" e B não diz nada.

56. A diz: "Nós dois somos cavaleiros" e B diz "A é um bandido".

57. A diz: "Eu sou um bandido ou B é um cavaleiro" e B não diz nada.

58. Ambos, A e B, dizem: "Eu sou um cavaleiro".

59. A diz: "Nós dois somos bandidos" e B não diz nada.

Os exercícios 60 a 65 são quebra-cabeças que podem ser resolvidos traduzindo as proposições em expressões lógicas e argumentos a partir destas expressões usando a tabela-verdade.

60. A polícia tem três suspeitos para o assassinado do sr. Cooper: sr. Smith, sr. Jones e sr. Williams. Smith, Jones e Williams declaram que não mataram Cooper. Smith também declara que Cooper era amigo de Jones e que Williams não gostava da vítima. Jones declara também que não conhecia Cooper e que estava fora da cidade no dia em que Cooper foi morto. Williams declara também que ele viu Smith e Jones com Cooper no dia em que ele foi morto e que ou Jones ou Smith o mataram. Você pode determinar quem foi o assassino se

a) um dos três é culpado e os dois inocentes estão falando a verdade, mas as declarações do homem culpado podem ser ou não falsas?

b) os homens inocentes não mentem?

61. Steve quer determinar os salários relativos de três colegas de trabalho usando dois fatos. Primeiro, ele sabe que se Fred não tem o maior salário dos três, então Janice tem. Segundo, ele sabe que se Janice não tem o salário mais baixo, então Maggie é a mais bem paga. É possível determinar os salários relativos de Fred, Maggie e Janice a partir do que Steve sabe? Se sim, quem tem o salário maior e quem tem o menor? Exponha seus argumentos.

62. Cinco amigos acessaram uma sala de bate-papo. É possível determinar quem está conversando se as seguintes informações são conhecidas? Kevin ou Heather, ou ambos, estão conversando. Randy ou Vijay, mas não ambos, estão conversando. Se Abby está conversando, então Randy também está. Vijay e Kevin estão ambos conversando, ou nenhum dos dois está. Se Heather está conversando, então estão também Abby e Kevin. Exponha seus argumentos.

63. Um detetive entrevistou quatro testemunhas de um crime. A partir das histórias das testemunhas, o detetive concluiu que, se o mordomo está dizendo a verdade, então o cozinheiro também está; o cozinheiro e o jardineiro, ambos, não podem estar dizendo a verdade; o jardineiro e o zelador, ambos, não estão mentindo; e se o zelador está dizendo a verdade, então o cozinheiro está mentindo. Para cada uma das quatro testemunhas, o detetive pode determinar se a pessoa está mentindo ou dizendo a verdade? Exponha seus argumentos.

64. Quatro amigos foram identificados como suspeitos de um acesso não autorizado em um sistema computacional. Eles fizeram declarações às autoridades que investigavam o crime. Alice disse: "Carlos que acessou". John disse "Eu não acessei". Carlos disse "Diana acessou". Diana disse "Carlos mentiu ao dizer que eu acessei".

a) Se as autoridades também sabem que apenas um dos quatro suspeitos está dizendo a verdade, quem cometeu o crime? Exponha seus argumentos.

b) Se as autoridades também sabem que apenas um dos quatro está mentindo, quem cometeu o crime? Exponha seus argumentos.

***65.** Resolva este famoso quebra-cabeça, atribuído a Albert Einstein e conhecido como o "**Enigma da Zebra**". Cinco homens de nacionalidades diferentes, com empregos diferentes, vivem em casas consecutivas em uma rua. Essas casas estão pintadas com cores diferentes. Os homens têm animais de estimação diferentes e gostam de bebidas diferentes. Determine quem tem uma zebra e quem tem por bebida favorita água mineral, a partir destas pistas: O inglês vive na casa vermelha. O espanhol tem um cachorro. O japonês é pintor. O italiano bebe chá. O norueguês vive na primeira casa à esquerda. A casa verde é imediatamente do lado direito da casa branca. O fotógrafo cria caracóis. O diplomata vive na casa amarela. Bebe-se leite na casa do meio. O dono da casa verde bebe café. A casa do norueguês é ao lado da azul. O violonista bebe suco de laranja. A raposa está na casa ao lado da do físico. O cavalo está na casa ao lado da do diplomata. [*Dica*: Faça uma tabela em que as filas representem os homens e as colunas, a cor das casas, seus empregos, seus animais e suas bebidas favoritas e use a argumentação lógica para determinar as entradas corretas na tabela.]

1.2 Equivalências Proposicionais

Introdução

Um importante tipo de passo usado na argumentação matemática é a substituição de uma proposição por outra com o mesmo valor-verdade. Por esse motivo, métodos que produzem proposições com o mesmo valor-verdade que uma dada proposição composta são usados largamente na construção de argumentos matemáticos. Note que vamos usar o termo "proposições compostas" para nos referir a uma expressão formada a partir de variáveis proposicionais que utilizam operadores lógicos, tais como $p \wedge q$.

Começaremos nossa discussão com a classificação de proposições compostas de acordo com seus possíveis valores-verdade.

DEFINIÇÃO 1

Uma proposição composta que é sempre verdadeira, qualquer que sejam os valores-verdade das proposições que ocorrem nela, é chamada de *tautologia*. Uma proposição composta que é sempre falsa, qualquer que seja o valor-verdade das proposições que a compõem, é chamada de *contradição*. Uma proposição composta que não é nem tautologia nem contradição é chamada de *contingência*.

Tautologias e contradições são muito importantes no raciocínio matemático. O Exemplo 1 ilustra esses tipos de proposições compostas.

TABELA 1 Exemplos de uma Tautologia e de uma Contradição.

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
V	F	V	F
F	V	V	F

TABELA 2 Leis de De Morgan.

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$
$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$

EXEMPLO 1 Podemos construir exemplos de tautologias e contradições usando apenas uma variável proposicional. Considere a tabela-verdade de $p \vee \neg p$ e $p \wedge \neg p$, mostrada na Tabela 1. Como $p \vee \neg p$ é sempre verdadeira, é uma tautologia. Como $p \wedge \neg p$ é sempre falsa, é uma contradição. ◀

Equivalências Lógicas

Demo

Proposições compostas que têm o mesmo valor-verdade em todos os possíveis casos são chamadas de **logicamente equivalentes**. Podemos definir esta noção como se segue.

DEFINIÇÃO 2 As proposições compostas p e q são chamadas de *logicamente equivalentes* se $p \leftrightarrow q$ é uma tautologia. A notação $p \equiv q$ indica que p e q são logicamente equivalentes.

Lembre-se: O símbolo $\equiv$ não é um conectivo lógico e $p \equiv q$ não é uma proposição composta, apenas quer dizer que $p \leftrightarrow q$ é uma tautologia. O símbolo $\Leftrightarrow$ é usado freqüentemente no lugar de $\equiv$ para indicar equivalências lógicas.

Exemplos Extras

Uma maneira de determinar quando duas proposições compostas são equivalentes é usar a tabela-verdade. Em particular, as proposições compostas p e q são equivalentes se e somente se as colunas que fornecem seus valores-verdade são idênticas. O Exemplo 2 ilustra esse método para estabelecer uma importantíssima e muito usada equivalência lógica: $\neg(p \vee q)$ é o mesmo que $\neg p \wedge \neg q$. Essa equivalência lógica é uma das duas **leis de De Morgan**, mostrada na Tabela 2, demonstradas pelo matemático inglês Augustus De Morgan, na metade do século XIX.

EXEMPLO 2 Mostre que $\neg(p \vee q)$ e $\neg p \wedge \neg q$ são logicamente equivalentes.

Solução: As tabelas-verdade dessas proposições compostas estão na Tabela 3. Como os valores-verdade $\neg(p \vee q)$ e $\neg p \wedge \neg q$ coincidem para todas as possibilidades de combinações de valores-verdade de p e q, segue-se que $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ é uma tautologia e, portanto, essas proposições compostas são logicamente equivalentes. ◀

TABELA 3 Tabelas-Verdade para $\neg(p \vee q)$ e $\neg p \wedge \neg q$.

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
V	V	V	F	F	F	F
V	F	V	F	F	V	F
F	V	V	F	V	F	F
F	F	F	V	V	V	V

TABELA 4 Tabela-Verdade para $\neg p \vee q$ e $p \rightarrow q$.

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
V	V	F	V	V
V	F	F	F	F
F	V	V	V	V
F	F	V	V	V

EXEMPLO 3 Mostre que $p \rightarrow q$ e $\neg p \vee q$ são logicamente equivalentes.

Solução: Construímos a tabela-verdade dessas proposições compostas na Tabela 4. Como os valores-verdade de $\neg p \vee q$ e $p \rightarrow q$ são idênticos, eles são logicamente equivalentes. ◀

Vamos agora estabelecer uma equivalência lógica entre duas proposições compostas que envolvem três variáveis proposicionais diferentes p, q e r. Para usar a tabela-verdade estabelecendo essa equivalência lógica, precisamos de oito linhas, uma para cada combinação de valores-verdade dessas três variáveis. Simbolicamente, nós representamos essas combinações listando os valores de p, q e r, respectivamente. Essas oito combinações de valores-verdade são VVV, VVF, VFV, VFF, FVV, FVF, FFV e FFF; usaremos essa ordem quando montarmos as linhas da tabela-verdade. Note que precisamos do dobro de linhas de que precisávamos quando tínhamos duas variáveis proposicionais; essa relação continua sendo válida para cada nova variável proposicional que venha a ser adicionada, então precisaremos de 16 linhas para estabelecer a equivalência entre duas proposições compostas com quatro variáveis proposicionais, e assim sucessivamente. Em geral, 2^n linhas são necessárias quando temos n variáveis proposicionais.

EXEMPLO 4 Mostre que $p \vee (q \wedge r)$ e $(p \vee q) \wedge (p \vee r)$ são logicamente equivalentes. Essa é a *propriedade distributiva* da disjunção sobre a conjunção.

Solução: Construímos a tabela-verdade para essas duas proposições compostas na Tabela 5. Como os valores-verdade de $p \vee (q \wedge r)$ e $(p \vee q) \wedge (p \vee r)$ são iguais, essas proposições são logicamente equivalentes. ◀

TABELA 5 Uma Demonstração de que $p \vee (q \wedge r)$ e $(p \vee q) \wedge (p \vee r)$ São Logicamente Equivalentes.

p	q	r	$q \wedge r$	$p \vee (q \wedge r)$	$p \vee q$	$p \vee r$	$(p \vee q) \wedge (p \vee r)$
V	V	V	V	V	V	V	V
V	V	F	F	V	V	V	V
V	F	V	F	V	V	V	V
V	F	F	F	V	V	V	V
F	V	V	V	V	V	V	V
F	V	F	F	F	V	F	F
F	F	V	F	F	F	V	F
F	F	F	F	F	F	F	F

TABELA 6 Equivalências Lógicas.	
Equivalências	***Nome***
$p \wedge \mathbf{V} \equiv p$ $p \vee \mathbf{F} \equiv p$	Propriedades dos elementos neutros
$p \vee \mathbf{V} \equiv \mathbf{V}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Propriedades de dominação
$p \vee p \equiv p$ $p \wedge p \equiv p$	Propriedades idempotentes
$\neg(\neg p) \equiv p$	Propriedade da dupla negação
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Propriedades comutativas
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Propriedades associativas
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Propriedades distributivas
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	Leis de De Morgan
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Propriedades de absorção
$p \vee \neg p \equiv \mathbf{V}$ $p \wedge \neg p \equiv \mathbf{F}$	Propriedades de negação

A Tabela 6 contém algumas equivalências importantes.* Nessas equivalências, **V** indica uma proposição composta que é sempre verdadeira, uma tautologia, e **F** indica uma proposição que é sempre falsa, uma contradição. Nós também mostramos algumas equivalências importantes que envolvem condicionais e bicondicionais nas tabelas 7 e 8, respectivamente. Ao leitor será pedido que verifique a veracidade dessas equivalências nos exercícios no final desta seção.

A propriedade associativa para a disjunção mostra que a expressão $p \vee q \vee r$ é bem definida, no sentido de que tanto faz qual disjunção é considerada primeiro, ou seja, tanto faz se fazemos primeiro $p \vee q$ e posteriormente a disjunção deste com r, ou se fazemos primeiro a disjunção de q com r e depois com p. De maneira análoga, $p \wedge q \wedge r$ também está bem definida. Estendendo esse raciocínio, segue-se que $p_1 \vee p_2 \vee \cdots \vee p_n$ e $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ também são bem definidas sempre que p_1, p_2, … p_n são proposições. Além disso, note que as leis de De Morgan podem ser estendidas para

$$\neg(p_1 \vee p_2 \vee \cdots \vee p_n) \equiv (\neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n)$$

e

$$\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \equiv (\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n).$$

(Métodos para demonstrar essas identidades serão analisados na Seção 4.1.)

---

* Leitores familiarizados com os conceitos de álgebra booleana vão notar que essas identidades são um caso especial de identidades que valem para qualquer álgebra booleana. Compare-as com o conjunto de identidades da Tabela 1 da Seção 2.2 e com as identidades booleanas da Tabela 5 na Seção 11.1.

TABELA 7 Equivalências Lógicas que Envolvem Sentenças Condicionais.
$p \to q \equiv \neg p \lor q$
$p \to q \equiv \neg q \to \neg p$
$p \lor q \equiv \neg p \to q$
$p \land q \equiv \neg(p \to \neg q)$
$\neg(p \to q) \equiv p \land \neg q$
$(p \to q) \land (p \to r) \equiv p \to (q \land r)$
$(p \to r) \land (q \to r) \equiv (p \lor q) \to r$
$(p \to q) \lor (p \to r) \equiv p \to (q \lor r)$
$(p \to r) \lor (q \to r) \equiv (p \land q) \to r$

TABELA 8 Equivalências Lógicas que Envolvem Bicondicionais.
$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \land q) \lor (\neg p \land \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Usando as Leis de De Morgan

As duas equivalências lógicas conhecidas como leis de De Morgan são particularmente importantes. Elas nos mostram como negar conjunções e como negar disjunções. Em particular, a equivalência $\neg(p \lor q) \equiv \neg p \land \neg q$ nos diz que a negação de uma disjunção é formada tomando a conjunção das negações das proposições componentes. Similarmente, $\neg(p \land q) \equiv \neg p \lor \neg q$ nos diz que a negação de uma conjunção é formada tomando a disjunção das negações das proposições componentes. O Exemplo 5 ilustra o uso das leis de De Morgan.

EXEMPLO 5 Use as leis de De Morgan para expressar as negações de "Miguel tem um celular e um laptop" e "Rodrigo vai ao concerto ou Carlos vai ao concerto".

Solução: Seja p "Miguel tem um celular" e q "Miguel tem um laptop". Então, "Miguel tem um celular e um laptop" pode ser representado por $p \land q$. Contudo, pela primeira lei de De Morgan, $\neg(p \land q)$ é equivalente a $\neg p \lor \neg q$. Conseqüentemente, podemos expressar a negação de nossa proposição original por "Miguel não tem um celular ou não tem um laptop".

Links

AUGUSTUS DE MORGAN (1806–1871) Augustus De Morgan nasceu na Índia, onde seu pai era coronel no exército indiano. A família De Morgan mudou-se para a Inglaterra quando ele tinha 7 meses de idade. Ele freqüentou escolas particulares, onde desenvolveu um grande interesse por matemática na sua juventude. De Morgan estudou na Universidade de Trinity, em Cambridge, graduando-se em 1827. Embora pensasse em entrar em medicina ou direito, De Morgan decidiu seguir carreira em matemática. Ele conquistou uma cadeira na Universidade de College, em Londres, em 1828, mas demitiu-se quando a faculdade despediu um colega sem apresentar as causas para a demissão. Entretanto, ele retomou essa cadeira em 1836, quando seu sucessor morreu, permanecendo até 1866.

De Morgan foi um professor notável que dava ênfase aos princípios mais que às técnicas. Entre seus estudantes estão muitos matemáticos famosos, incluindo Augusta Ada, Condessa de Lovelace, que era colaboradora de Charles Babbage em seu trabalho com máquinas computacionais (veja a página 27 nas notas biográficas de Augusta Ada). (De Morgan preveniu a condessa de que estudar matemática em excesso, poderia interferir em suas habilidades maternais!)

De Morgan foi um escritor extremamente prolífico. Escreveu milhares de artigos para mais de 15 periódicos. Também escreveu livros teóricos sobre muitos assuntos, incluindo lógica, probabilidade, cálculo e álgebra. Em 1838, ele apresentou o que talvez tenha sido a primeira explicação clara de uma importante técnica de demonstração, conhecida como *indução matemática* (discutida na Seção 4.1 deste livro), termo que ele cunhou. Na década de 1840, De Morgan fez contribuições fundamentais ao desenvolvimento da lógica simbólica. Ele criou notações que o ajudaram a provar equivalências proposicionais, assim como as leis que receberam seu nome. Em 1842, De Morgan apresentou o que talvez tenha sido a primeira definição precisa de limite e o desenvolvimento de alguns testes de convergência de séries infinitas. De Morgan interessou-se também pela história da matemática, escrevendo biografias de Newton e Halley.

Em 1837, De Morgan casou-se com Sophia Frend, que escreveu a biografia do marido em 1882. A pesquisa, a escrita e o ensino de De Morgan deixaram pouco tempo para ele se dedicar a sua família e vida social. No entanto, ele ficou conhecido pela sua bondade, bom humor e grande inteligência.

Seja r "Rodrigo vai ao concerto" e s "Carlos vai ao concerto". Então, "Rodrigo vai ao concerto ou Carlos vai ao concerto" pode ser representado por $r \vee s$. E, pela segunda lei de De Morgan, temos que $\neg(r \vee s)$ é equivalente a $\neg r \wedge \neg s$. Logo, podemos expressar sua negação por "Rodrigo não vai ao concerto e Carlos não vai ao concerto". ◀

Construindo Novas Equivalências Lógicas

As equivalências lógicas na Tabela 6, assim como qualquer outra que seja estabelecida (como as mostradas nas tabelas 7 e 8), podem ser usadas para construir equivalências lógicas adicionais. A razão para isso é que uma proposição composta pode ser substituída por outra proposição composta que é logicamente equivalente a essa sem mudar o valor-verdade da proposição original. Essa técnica é ilustrada nos exemplos 6–8, em que também usamos o fato de que se p e q são logicamente equivalentes e q e r são logicamente equivalentes, então p e r também são logicamente equivalentes (veja o Exercício 56).

EXEMPLO 6 Mostre que $\neg(p \rightarrow q)$ e $p \wedge \neg q$ são logicamente equivalentes.

Exemplos Extras

Solução: Podemos usar uma tabela-verdade para mostrar que essas proposições compostas são equivalentes (como no Exemplo 4). Inclusive, não deve ser difícil fazer isso. No entanto, queremos ilustrar como usar identidades lógicas que já conhecemos para estabelecer novas identidades lógicas, isso porque esse método tem uma importância prática para estabelecer equivalências de proposições compostas com um grande número de variáveis. Então, vamos estabelecer essa equivalência desenvolvendo uma série de equivalências lógicas, usando uma das equivalências da Tabela 6 por vez, começando por $\neg(p \rightarrow q)$ e terminando com $p \wedge \neg q$. Temos, assim, as equivalências a seguir.

$$\begin{aligned} \neg(p \rightarrow q) &\equiv \neg(\neg p \vee q) && \text{pelo Exemplo 3} \\ &\equiv \neg(\neg p) \wedge \neg q && \text{pela segunda lei de De Morgan} \\ &\equiv p \wedge \neg q && \text{pela propriedade da dupla negação} \end{aligned}$$ ◀

EXEMPLO 7 Mostre que $\neg(p \vee (\neg p \wedge q))$ e $\neg p \wedge \neg q$ são logicamente equivalentes desenvolvendo uma série de equivalências lógicas.

Solução: Vamos usar uma das equivalências da Tabela 6 por vez, começando por $\neg(p \vee (\neg p \wedge q))$ e terminando com $\neg p \wedge \neg q$. (*Nota:* Poderíamos estabelecer essa equivalência facilmente usando tabelas-verdade.) Assim, temos as equivalências a seguir.

$$\begin{aligned} \neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) && \text{pela segunda lei de De Morgan} \\ &\equiv \neg p \wedge [\neg(\neg p) \vee \neg q] && \text{pela primeira lei de De Morgan} \\ &\equiv \neg p \wedge (p \vee \neg q) && \text{pela propriedade da dupla negação} \\ &\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{pela segunda propriedade distributiva} \\ &\equiv \mathbf{F} \vee (\neg p \wedge \neg q) && \text{pois } \neg p \wedge p \equiv \mathbf{F} \\ &\equiv (\neg p \wedge \neg q) \vee \mathbf{F} && \text{pela propriedade comutativa para disjunções} \\ &\equiv \neg p \wedge \neg q && \text{pela propriedade dos elementos neutros para } \mathbf{F} \end{aligned}$$

Em conseqüência, $\neg(p \vee (\neg p \wedge q))$ e $\neg p \wedge \neg q$ são logicamente equivalentes. ◀

EXEMPLO 8 Mostre que $(p \wedge q) \rightarrow (p \vee q)$ é uma tautologia.

Solução: Para mostrar que essa proposição é uma tautologia, vamos usar equivalências para demonstrar que é logicamente equivalente a **V**. (*Nota:* Isso poderia ser feito usando tabelas-verdade.)

$$
\begin{aligned}
(p \wedge q) \rightarrow (p \vee q) &\equiv \neg(p \wedge q) \vee (p \vee q) && \text{pelo Exemplo 3}\\
&\equiv (\neg p \vee \neg q) \vee (p \vee q) && \text{pela primeira lei de De Morgan}\\
&\equiv (\neg p \vee p) \vee (\neg q \vee q) && \text{pelas propriedades associativas e comutativas para a disjunção}\\
&\equiv \mathbf{V} \vee \mathbf{V} && \text{pelo Exemplo 1 e pela propriedade comutativa para a disjunção}\\
&\equiv \mathbf{V} && \text{pela propriedade de dominação}
\end{aligned}
$$

◀

Uma tabela-verdade pode ser usada para determinar se uma proposição composta é uma tautologia. Isso pode ser feito rapidamente se for uma proposição composta com poucas variáveis, mas, quando o número de variáveis cresce, isso pode ficar impraticável. Por exemplo, existem $2^{20} = 1.048.576$ linhas em uma tabela-verdade para uma proposição composta com 20 variáveis proposicionais. Claramente você precisará de um computador para ajudá-lo a determinar quando uma proposição composta é uma tautologia. Quando, no entanto, existem 1.000 variáveis proposicionais, um computador pode determinar em um tempo razoável se uma proposição é uma tautologia? Testando todas as $2^{1.000}$ (um número com mais de 300 algarismos decimais) possíveis combinações de valores-verdade, um computador não consegue terminar em menos de alguns trilhões de anos. Além disso, não existe um outro método conhecido que um computador possa seguir para determinar em um tempo razoável quando uma proposição com muitas variáveis proposicionais é uma tautologia. Vamos estudar questões como essas no Capítulo 3, quando estudaremos a complexidade de algoritmos.

Links

AUGUSTA ADA, CONDESSA DE LOVELACE (1815–1852) Augusta Ada foi a única filha do casamento do famoso poeta Lorde Byron e Lady Byron, Annabella Millbanke, que se separaram quando Ada tinha 1 mês de idade, por causa do escândalo amoroso de Lorde Byron com sua meia-irmã. Lorde Byron tinha uma reputação, descrita por uma de suas amantes como "louco, mal e perigoso". Lady Byron era notável por sua inteligência e tinha paixão por matemática; ela era chamada por Lorde Byron de "A Princesa dos Paralelogramos". Augusta foi criada por sua mãe, que encorajou seus talentos intelectuais, especialmente na música e na matemática, tendo em vista que considerava perigosas as tendências poéticas. Naquela época, não era permitido que as mulheres freqüentassem as universidades nem se juntassem a grupos de estudos. No entanto, Augusta adquiriu seus estudos matemáticos sozinha e com matemáticos, incluindo William Frend. Ela também tinha o apoio de outra matemática, Mary Somerville, e, em 1834, em um jantar na casa de Mary Somerville, ela foi apresentada às idéias de Charles Babbage sobre uma máquina de calcular, chamada "Engenho Analítico". Em 1838, Augusta Ada casou-se com Lorde King, elevado posteriormente a Conde de Lovelace. Juntos, eles tiveram três filhos.

Augusta Ada continuou seus estudos em matemática depois do casamento. Charles Babbage continuou trabalhando em seu "Engenho Analítico" e apresentando-o para a Europa. Em 1842, Babbage pediu a Augusta Ada que traduzisse um artigo para o francês, descrevendo sua invenção. Quando Babbage viu a tradução, sugeriu que ela começasse a escrever suas próprias anotações, e o resultado final foi três vezes o original. Os relatos mais completos sobre a máquina de Babbage estão nas anotações de Augusta Ada. Em suas anotações, ela comparou o trabalho do "Engenho Analítico" ao tear de Jacquard, com a analogia dos cartões perfurados de Babbage aos usados para criar estampas no tear. Além disso, ela reconheceu a promessa da máquina como uma proposta de computador muito melhor do que fez Babbage. Ela constatou que "o motor é a expressão material de qualquer função indefinida de qualquer grau de generalidade e complexidade". Suas anotações sobre o "Engenho Analítico" anteciparam futuros desenvolvimentos. Augusta Ada publicou seus escritos sob as iniciais. A.A.L. para ocultar sua identidade como mulher, assim como muitas mulheres fizeram naquele tempo em que não eram consideradas intelectuais como os homens. Depois de 1845, ela e Babbage trabalharam juntos no desenvolvimento de um sistema para determinar raças de cavalos. Infelizmente esse sistema não funcionou muito bem, deixando Augusta extremamente debilitada fisicamente, contraindo câncer de útero ainda muito jovem.

Em 1953, as anotações de Augusta Ada sobre o "Engenho Analítico" foram republicadas, 100 anos após a sua escrita e depois de muito tempo esquecidas. Em seu trabalho, na década de 1950, sobre a capacidade de os computadores pensarem (e seu famoso teste Turing), Alan Turing respondeu à declaração de Augusta Ada de que "o Engenho Analítico não tem a pretensão de dar origem a nada. Ele pode fazer o que conhecemos para organizar sua performance". Esse "diálogo" entre Turing e Augusta Ada é ainda assunto de controvérsias. Por causa de suas contribuições fundamentais à computação, a linguagem computacional "Augusta" recebeu esse nome em homenagem à Condessa de Lovelace.

Exercícios

1. Use a tabela-verdade para verificar estas equivalências.
a) $p \wedge \mathbf{V} \equiv p$ **b)** $p \vee \mathbf{F} \equiv p$
c) $p \wedge \mathbf{F} \equiv \mathbf{F}$ **d)** $p \vee \mathbf{V} \equiv \mathbf{V}$
e) $p \vee p \equiv p$ **f)** $p \wedge p \equiv p$

2. Mostre que $\neg(\neg p)$ e p são logicamente equivalentes.

3. Use a tabela-verdade para verificar as propriedades comutativas.
a) $p \vee q \equiv q \vee p$ **b)** $p \wedge q \equiv q \wedge p$

4. Use a tabela-verdade para verificar as propriedades associativas.
a) $(p \vee q) \vee r \equiv p \vee (q \vee r)$
b) $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

5. Use a tabela-verdade para verificar a propriedade distributiva.
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$.

6. Use a tabela-verdade para verificar a primeira lei de De Morgan.
$\neg(p \wedge q) \equiv \neg p \vee \neg q$.

7. Use as leis de De Morgan para encontrar a negação de cada uma das proposições abaixo.
a) Jan é rica e feliz.
b) Carlos andará de bicicleta ou correrá amanhã.
c) Mei anda ou pega o ônibus para ir à escola.
d) Ibrahim é esperto e trabalha muito.

8. Use as leis de De Morgan para encontrar a negação de cada uma das proposições abaixo.
a) Kwame trabalhará na indústria ou irá para a faculdade.
b) Yoshiko conhece Java e cálculo.
c) James é jovem e forte.
d) Rita mudará para Oregon ou Washington.

9. Mostre que cada uma das proposições condicionais a seguir é uma tautologia, usando a tabela-verdade.
a) $(p \wedge q) \rightarrow p$ **b)** $p \rightarrow (p \vee q)$
c) $\neg p \rightarrow (p \rightarrow q)$ **d)** $(p \wedge q) \rightarrow (p \rightarrow q)$
e) $\neg(p \rightarrow q) \rightarrow p$ **f)** $\neg(p \rightarrow q) \rightarrow \neg q$

10. Mostre que cada uma das proposições condicionais abaixo é uma tautologia, usando a tabela-verdade.
a) $[\neg p \wedge (p \vee q)] \rightarrow q$
b) $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$
c) $[p \wedge (p \rightarrow q)] \rightarrow q$
d) $[(p \vee q) \wedge (p \rightarrow r) \wedge (q \rightarrow r)] \rightarrow r$

11. Mostre que cada uma das proposições condicionais do Exercício 9 é uma tautologia, sem usar a tabela-verdade.

12. Mostre que cada uma das proposições condicionais do Exercício 10 é uma tautologia, sem usar a tabela-verdade.

13. Use a tabela-verdade para verificar as propriedades de absorção.
a) $p \vee (p \wedge q) \equiv p$ **b)** $p \wedge (p \vee q) \equiv p$

14. Determine se $(\neg p \wedge (p \rightarrow q)) \rightarrow \neg\, q$ é uma tautologia.

15. Determine se $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg\, p$ é uma tautologia.

Os exercícios 16 a 28 pedem que você mostre que duas proposições compostas são logicamente equivalentes. Para fazer isso, ou mostre que os dois lados são verdadeiros, ou que os dois são falsos, para exatamente as mesmas combinações de valores-verdade das variáveis proposicionais nessas expressões (o que for mais fácil).

16. Mostre que $p \leftrightarrow q$ e $(p \wedge q) \vee (\neg p \wedge \neg q)$ são equivalentes.

17. Mostre que $\neg(p \leftrightarrow q)$ e $p \leftrightarrow \neg q$ são logicamente equivalentes.

HENRY MAURICE SHEFFER (1883–1964) Henry Maurice Sheffer, filho de pais judeus, nasceu no oeste da Ucrânia e emigrou para os Estados Unidos em 1892 com seus pais e seis irmãos. Estudou na Escola Latina de Boston antes de entrar em Harvard, onde completou sua graduação em 1905, seu mestrado em 1907 e seu Ph.D. em filosofia em 1908. Depois de conquistar uma posição de pós-doutor em Harvard, Henry viajou para a Europa com bolsa de pesquisa. Ao retornar para os Estados Unidos, ele se tornou um acadêmico nômade, permanecendo um ano em cada universidade: Universidade de Washington, Cornell, Minessota, Missouri e Universidade da Cidade, em Nova York. Em 1916, ele retornou a Harvard como membro do corpo docente do departamento de filosofia. Permaneceu em Harvard até aposentar-se, em 1952.

Sheffer introduziu, em 1913, o que conhecemos hoje por "golpe de Sheffer" que se tornou famoso apenas depois que foi usado em 1925 na edição de *Principia Mathematica,* de Whitehead e Russell. Nessa mesma edição, Russell escreveu que Sheffer tinha inventado um poderoso método que poderia ser usado para simplificar a *Principia*. Por causa desse comentário, Sheffer era visto como um homem misterioso para os lógicos, especialmente porque ele, que teve poucas publicações ao longo de sua carreira, nunca publicou os detalhes desse método, que foi apenas descrito em notas de mimiógrafo e em uma breve publicação abstrata.

Sheffer foi um professor dedicado de lógica matemática. Ele gostava de ministrar aulas em turmas pequenas; não gostava de auditórios. Quando estranhos apareciam em suas aulas, Sheffer pedia-lhes que se retirassem, mesmo se fossem colegas ou convidados que iam visitar Harvard. Sheffer tinha apenas um metro e meio de altura; era notado por sua inteligência e vigor, assim como pelo seu nervosismo e irritabilidade. Embora muito inteligente, ele era muito sozinho. Ele é conhecido pela piada que fez ao aposentar-se: "Professores velhos nunca morrem, tornam-se eméritos". Sheffer também tem o crédito de cunhar a expressão "álgebra booleana" (assunto do Capítulo 11 deste livro). Ele foi casado por um curto espaço de tempo e viveu a maior parte de sua vida madura em um quarto de hotel, com seus livros de lógica e um vasto arquivo de papéis em que ele costumava anotar suas idéias. Infelizmente, Sheffer sofreu de depressão profunda durante as duas últimas décadas de sua vida.

18. Mostre que $p \rightarrow q$ e $\neg q \rightarrow \neg p$ são logicamente equivalentes.

19. Mostre que $\neg p \leftrightarrow q$ e $p \leftrightarrow \neg q$ são logicamente equivalentes.

20. Mostre que $\neg(p \oplus q)$ e $p \leftrightarrow q$ são logicamente equivalentes.

21. Mostre que $\neg(p \leftrightarrow q)$ e $\neg p \leftrightarrow q$ são logicamente equivalentes.

22. Mostre que $(p \rightarrow q) \wedge (p \rightarrow r)$ e $p \rightarrow (q \wedge r)$ são logicamente equivalentes.

23. Mostre que $(p \rightarrow r) \wedge (q \rightarrow r)$ e $(p \vee q) \rightarrow r$ são logicamente equivalentes.

24. Mostre que $(p \rightarrow q) \vee (p \rightarrow r)$ e $p \rightarrow (q \vee r)$ são logicamente equivalentes.

25. Mostre que $(p \rightarrow r) \vee (q \rightarrow r)$ e $(p \wedge q) \rightarrow r$ são logicamente equivalentes.

26. Mostre que $\neg p \rightarrow (q \rightarrow r)$ e $q \rightarrow (p \vee r)$ são logicamente equivalentes.

27. Mostre que $p \leftrightarrow q$ e $(p \rightarrow q) \wedge (q \rightarrow p)$ são logicamente equivalentes.

28. Mostre que $p \leftrightarrow q$ e $\neg p \leftrightarrow \neg q$ são logicamente equivalentes.

29. Mostre que $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$ é uma tautologia.

30. Mostre que $(p \vee q) \wedge (\neg p \vee r) \rightarrow (q \vee r)$ é uma tautologia.

31. Mostre que $(p \rightarrow q) \rightarrow r$ e $p \rightarrow (q \rightarrow r)$ não são equivalentes.

32. Mostre que $(p \wedge q) \rightarrow r$ e $(p \rightarrow r) \wedge (q \rightarrow r)$ não são equivalentes.

33. Mostre que $(p \rightarrow q) \rightarrow (r \rightarrow s)$ e $(p \rightarrow r) \rightarrow (q \rightarrow s)$ não são logicamente equivalentes.

O **dual** de uma proposição composta que contém apenas operadores lógicos $\vee$, $\wedge$ e $\neg$ é a proposição composta obtida pela troca de cada $\vee$ por $\wedge$, cada $\wedge$ por $\vee$, cada **V** por **F** e cada **F** por **V**. O dual de s é representado por s^*.

34. Encontre o dual de cada uma destas proposições compostas.

a) $p \vee \neg q$ **b)** $p \wedge (q \vee (r \wedge \mathbf{V}))$

c) $(p \wedge \neg q) \vee (q \wedge \mathbf{F})$

35. Encontre o dual de cada uma destas proposições compostas.

a) $p \wedge \neg q \wedge \neg r$ **b)** $(p \wedge q \wedge r) \vee s$

c) $(p \vee \mathbf{F}) \wedge (q \vee \mathbf{V})$

36. Quando $s^* = s$, em que s é uma proposição composta?

37. Mostre que $(s^*)^* = s$ quando s é uma proposição composta.

38. Mostre que as equivalências lógicas da Tabela 6, exceto pela propriedade da dupla negação, vêm em pares, em que cada par contém proposições compostas que são duais de si próprias.

****39.** Por que os duais de duas proposições compostas equivalentes são também equivalentes, em que essas proposições compostas contêm apenas os operadores $\wedge$, $\vee$ e $\neg$?

40. Encontre uma proposição composta que envolva as variáveis proposicionais p, q e r, que é verdadeira quando p e q são verdadeiras e r é falsa, mas o contrário é falso. [*Dica*: Use uma conjunção de cada variável proposicional ou sua negação.]

41. Encontre uma proposição composta que envolva as variáveis proposicionais p, q e r, que é verdadeira quando exatamente duas de p, q e r forem verdadeiras, mas o contrário é falso. [*Dica:* Forme uma disjunção de conjunções. Inclua uma conjunção para cada combinação de valores para os quais a variante proposicional for verdadeira. Cada conjunção deverá incluir cada uma dessas três variáveis ou suas negações.]

☞**42.** Suponha que a tabela-verdade em n variáveis proposicionais é dada. Mostre que pode ser formada uma proposição composta com essa tabela-verdade a partir de uma disjunção das conjunções das variáveis, ou suas negações, com uma conjunção formada por cada combinação de valores para os quais a proposição composta é verdadeira. A proposição composta resultante é chamada de **forma normal disjuntiva**.

Um conjunto de operadores lógicos é chamado de **funcionalmente completo** quando todas as proposições compostas são logicamente equivalentes a uma proposição composta que envolva apenas esses operadores lógicos.

43. Mostre que $\neg$, $\wedge$ e $\vee$ formam um grupo de operadores lógicos funcionalmente completo. [*Dica:* Use o fato de que toda proposição composta é logicamente equivalente a outra em uma forma normal disjuntiva, como visto no Exercício 42.]

***44.** Mostre que $\neg$ e $\wedge$ formam um grupo de operadores lógicos funcionalmente completo. [*Dica:* Primeiro use a lei de De Morgan para mostrar que $p \vee q$ é equivalente a $\neg(\neg p \wedge \neg q)$.]

***45.** Mostre que $\neg$ e $\vee$ formam um grupo de operadores lógicos funcionalmente completo.

Os exercícios subseqüentes envolvem os operadores lógicos *NAND* e *NOR*. A proposição p *NAND* q é verdadeira quando ou p ou q, ou ambas, forem falsas; e é falsa quando p e q, ambas, forem verdadeiras. A proposição p *NOR* q é verdadeira quando ambas, p e q, forem falsas, e é falsa em qualquer outro caso. As proposições p *NAND* q e p *NOR* q são indicadas por $p \mid q$ e $p \downarrow q$, respectivamente. (Os operadores $\mid$ e $\downarrow$ são chamados de **conectivo de Sheffer** e **flecha de Peirce**, recebendo os nomes de H. M. Sheffer e C. S. Peirce, respectivamente.)

46. Construa a tabela-verdade para o operador lógico *NAND*.

47. Mostre que $p \mid q$ é logicamente equivalente a $\neg(p \wedge q)$.

48. Construa a tabela-verdade para o operador lógico *NOR*.

49. Mostre que $p \downarrow q$ é logicamente equivalente a $\neg(p \vee q)$.

50. Neste exercício, mostraremos que $\{\downarrow\}$ é um conjunto de operadores lógicos funcionalmente completo.

a) Mostre que $p \downarrow p$ é logicamente equivalente a $\neg p$.

b) Mostre que $(p \downarrow q) \downarrow (p \downarrow q)$ é logicamente equivalente a $p \vee q$.

c) Conclua a partir dos itens (a) e (b) e do Exercício 49 que $\{\downarrow\}$ é um conjunto de operadores lógicos funcionalmente completo.

***51.** Encontre uma proposição composta logicamente equivalente a $p \rightarrow q$, usando apenas o operador lógico $\downarrow$.

52. Mostre que $\{\mid\}$ é um conjunto de operadores lógicos funcionalmente completo.

53. Mostre que $p \mid q$ e $q \mid p$ são equivalentes.

54. Mostre que $p \mid (q \mid r)$ e $(p \mid q) \mid r$ não são equivalentes; portanto, o operador lógico $\mid$ não é associativo.

***55.** Quantas formas diferentes de tabelas-verdade de proposições compostas existem que envolvam as variantes proposicionais p e q?

56. Mostre que se p, q e r são proposições compostas, em que p e q são logicamente equivalentes e q e r são também logicamente equivalentes, então p e r são logicamente equivalentes.

57. A sentença a seguir foi tirada das especificações de um sistema telefônico: "Se o diretório de dados for do banco aberto, então o monitor é posto em estado de fechamento, se o sistema não estiver em seu estado inicial". Essa especificação é difícil de ser compreendida porque envolve proposições com duas condicionais. Encontre um equivalente, uma especificação de fácil compreensão, que envolva disjunções e negações, mas não proposições condicionais.

58. Quantas das disjunções $p \vee \neg q$, $\neg p \vee q$, $q \vee r$, $q \vee \neg r$ e $\neg q \vee \neg r$ podem ser verdadeiras simultaneamente, a partir da construção de uma tabela-verdade com valores para p, q e r?

59. Quantas das disjunções $p \vee \neg q \vee s$, $\neg p \vee \neg r \vee s$, $\neg p \vee \neg r \vee \neg s$, $\neg p \vee q \vee \neg s$, $q \vee r \vee \neg s$, $q \vee \neg r \vee \neg s$, $\neg p \vee \neg q \vee \neg s$, $p \vee r \vee s$ e $p \vee r \vee \neg s$ podem ser verdadeiras simultaneamente, a partir da construção de uma tabela-verdade com valores para p, q, r e s?

Uma proposição composta é **satisfatória** se existe uma atribuição de valores-verdade para as variáveis na proposição que torna a declaração verdadeira.

60. Quais das proposições compostas abaixo são satisfatórias?

a) $(p \vee q \vee \neg r) \wedge (p \vee \neg q \vee \neg s) \wedge (p \vee \neg r \vee \neg s) \wedge (\neg p \vee \neg q \vee \neg s) \wedge (p \vee q \vee \neg s)$

b) $(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg s) \wedge (p \vee \neg q \vee \neg s) \wedge (\neg p \vee \neg r \vee \neg s) \wedge (p \vee q \vee \neg r) \wedge (p \vee \neg r \vee \neg s)$

c) $(p \vee q \vee r) \wedge (p \vee \neg q \vee \neg s) \wedge (q \vee \neg r \vee s) \wedge (\neg p \vee r \vee s) \wedge (\neg p \vee q \vee \neg s) \wedge (p \vee \neg q \vee \neg r) \wedge (\neg p \vee \neg q \vee s) \wedge (\neg p \vee \neg r \vee \neg s)$

61. Explique como um algoritmo para definir se uma proposição composta é satisfatória pode ser usado para determinar se uma proposição composta é uma tautologia. [*Dica:* Observe $\neg p$, em que p é a proposição composta que está sendo examinada.]

1.3 Predicados e Quantificadores

Introdução

A lógica proposicional, estudada nas seções 1.1 e 1.2, não pode expressar adequadamente o significado das proposições em matemática e em linguagem natural. Por exemplo, suponha que saibamos que

"Todo computador conectado à rede da universidade está funcionando apropriadamente."

Nenhuma regra da lógica proposicional nos permite decidir sobre a veracidade da afirmação

"MATH3 está funcionando apropriadamente,"

em que MATH3 é um dos computadores conectados à rede da universidade. Da mesma forma, não podemos usar as regras da lógica proposicional para concluir da proposição

"CS2 está sob ataque de um hacker."

em que CS2 é um computador na rede da universidade, para concluir que é verdade que

"Existe um computador na rede da universidade que está sob ataque de um hacker."

Nesta seção, vamos introduzir uma lógica mais poderosa chamada **lógica de predicados**. Veremos como a lógica de predicados pode ser usada para expressar o significado de um amplo grupo de proposições em matemática e em ciência da computação de modo que nos permita raciocinar e explorar relações entre objetos. Para entender a lógica de predicados, precisamos primeiramente introduzir o conceito de predicado. Posteriormente, vamos introduzir o conceito de quantificadores, que nos permite raciocinar com declarações sobre determinada propriedade que vale para todos os objetos de certo tipo e com declarações sobre a existência de um objeto com uma propriedade específica.

Predicados

Sentenças que envolvem variáveis, tais como

"$x > 3$", "$x = y + 3$", "$x + y = z$",

"computador x está sob ataque de um hacker"

e

"computador x está funcionando apropriadamente",

são freqüentemente encontradas na matemática, em programas de computador e em sistemas de especificações. Essas declarações não são nem verdadeiras nem falsas quando o valor das variáveis não é especificado. Nesta seção, vamos discutir como proposições podem ser produzidas a partir dessas declarações.

A declaração "x é maior que 3" tem duas partes. A primeira, a variável x, é o sujeito da declaração. A segunda — o **predicado**, "é maior que 3" — refere-se a uma propriedade que o sujeito pode ter. Podemos representar a declaração "x é maior que 3" por $P(x)$, em que P indica o predicado "é maior que 3" e x é a variável. A declaração, ou afirmação, é também chamada de o valor da **função proposicional** P em x. Uma vez que um valor é dado para a variável x, a declaração $P(x)$ torna-se uma proposição e tem um valor-verdade. Considere os exemplos 1 e 2.

EXEMPLO 1 Seja $P(x)$ a declaração "$x > 3$". Qual o valor-verdade de $P(4)$ e $P(2)$?

Solução: Obtemos a proposição $P(4)$ substituindo $x = 4$ na declaração "$x > 3$". Então, $P(4)$, que é a proposição "$4 > 3$", é verdadeira. No entanto, $P(2)$, que é a proposição "$2 > 3$", é falsa. ◀

EXEMPLO 2 Seja $A(x)$ a declaração "O computador x está sendo invadido por um hacker". Suponha que dos computadores do campus apenas o CS2 e o MATH1 estão sendo invadidos por algum hacker. Quais os valores-verdade de $A(\text{CS1})$, $A(\text{CS2})$ e $A(\text{MATH1})$?

Solução: Obtemos a proposição $A(\text{CS1})$ substituindo x por CS1 na declaração "O computador x está sendo invadido por um hacker". Como CS1 não está na lista dos computadores invadidos, concluímos que $A(\text{CS1})$ é falsa. De maneira similar, como CS2 e MATH1 estão na lista dos invadidos, sabemos que $A(\text{CS2})$ e $A(\text{MATH1})$ são verdadeiras. ◀

Também podemos trabalhar com afirmações que envolvam mais que uma variável. Por exemplo, considere a afirmação "$x = y + 3$". Podemos indicá-la por $Q(x, y)$, em que x e y são variáveis e Q é o predicado. Quando estabelecemos valores para as variáveis, a proposição $Q(x, y)$ tem um valor-verdade.

EXEMPLO 3 Seja $Q(x, y)$ a representação de "$x = y + 3$". Quais os valores-verdade de $Q(1, 2)$ e $Q(3, 0)$?

Exemplos Extras

Solução: Para obter $Q(1, 2)$, basta tomar $x = 1$ e $y = 2$ na equação representada por $Q(x, y)$. Portanto, $Q(1, 2)$ é a proposição "$1 = 2 + 3$", que é falsa. A afirmação $Q(3, 0)$ é a proposição "$3 = 0 + 3$", que é verdadeira. ◀

EXEMPLO 4 Seja $A(c, n)$ a representação de "O computador c está conectado à rede n", em que c é uma variável que indica computadores e n é uma variável que indica redes. Suponha que o computador MATH1 está conectado à rede CAMPUS2, mas não à rede CAMPUS1. Quais os valores-verdade de A(MATH1, CAMPUS1) e A(MATH1, CAMPUS2)?

Solução: Como MATH1 não está conectado à rede CAMPUS1, vemos que A(MATH1, CAMPUS1) é falsa. Por outro lado, MATH1 está conectado à rede CAMPUS2, logo A(MATH1, CAMPUS2) é verdadeira. ◀

De maneira análoga, podemos tomar afirmações com três variáveis, como $R(x, y, z)$ representando "$x + y = z$". Quando valores são atribuídos às variáveis, a proposição derivada tem um valor-verdade.

EXEMPLO 5 Quais os valores-verdade para $R(1, 2, 3)$ e $R(0, 0, 1)$?

Solução: A proposição $R(1, 2, 3)$ é obtida substituindo-se $x = 1$, $y = 2$ e $z = 3$ na declaração $R(x, y, z)$. Então, vemos que $R(1, 2, 3)$ representa "$1 + 2 = 3$", que é verdadeira. Também podemos notar que $R(0, 0, 1)$ representa "$0 + 0 = 1$", que é falsa. ◀

Em geral, uma afirmação que envolva n variáveis $x_1, x_2, \ldots, x_n$ pode ser indicada por

$$P(x_1, x_2, \ldots, x_n).$$

A declaração, ou afirmação, indicada por $P(x_1, x_2, \ldots, x_n)$ é chamada de valor da **função proposicional** P para a n-úpla $(x_1, x_2, \ldots, x_n)$, e P é chamado de **predicado *n*-ário**.

Funções proposicionais ocorrem em programas de computação, como mostrado no Exemplo 6.

Links

CHARLES SANDERS PEIRCE (1839–1914) Muitos consideram Charles Peirce o intelectual mais original e versátil dos Estados Unidos; ele nasceu em Cambridge, Massachusetts, e fez importantes contribuições em um grande número de disciplinas, incluindo matemática, astronomia, química, geodésica, metrologia, engenharia, psicologia, filologia, história da ciência e economia. Charles era também inventor, estudante de medicina dedicado, revisor de livros, dramaturgo e ator, escritor de contos, fenomenologista, lógico e metafísico. Ele ficou conhecido pela sua competência filosófica construtivista e produtividade em lógica, matemática e muitas outras áreas da ciência. Seu pai, Benjamin Peirce, era professor de matemática e filosofia natural de Harvard. Peirce freqüentou Harvard (1855–1859) e recebeu seu diploma de mestrado em artes (1862) e um diploma de doutorado em química pela Escola Científica Lawrence (1863). Seu pai o apoiou a seguir a carreira científica, mas, em vez disso, ele escolheu estudar lógica e metodologia científica.

Em 1861, Peirce se tornou um membro da Agrimensura da Costa dos Estados Unidos, com o objetivo de melhor compreender a metodologia científica. Seus serviços para a Agrimensura o dispensaram dos serviços militares durante a Guerra Civil. Enquanto trabalhava para a Agrimensura, Peirce deu continuidade a seus trabalhos nas áreas de astronomia e geodésica. Ele deu contribuições fundamentais na criação de pêndulos e projetos de mapas, aplicando novos desenvolvimentos matemáticos na teoria de funções elípticas. Ele foi a primeira pessoa a usar ondas de luz como unidade de medida. Peirce foi promovido a Assistente na Agrimensura, posição em que se manteve até que foi obrigado a largá-la em 1891, quando ele não concordou com a direção tomada pela administração da Agrimensura.

Embora tenha dedicado a maior parte do tempo às ciências físicas, Peirce desenvolveu uma hierarquia de ciências, com a matemática em seu topo, no qual os métodos de uma ciência poderiam ser adaptados para serem usados pelas ciências que estivessem abaixo na hierarquia. Ele foi também o fundador da teoria filosófica americana de pragmatismo.

A única posição acadêmica que Peirce conquistou foi a de mestre em lógica na Universidade John Hopkins, em Baltimore, de 1879 a 1884. Seu trabalho matemático durante esse período inclui contribuições à lógica, teoria dos conjuntos, álgebra abstrata e filosofia da matemática. Seu trabalho é relevante até nos dias de hoje; alguns de seus trabalhos em lógica foram recentemente aplicados à inteligência artificial. Peirce acreditava que o estudo da matemática poderia desenvolver o poder mental da imaginação, abstração e generalização. Suas diversas atividades, depois de aposentar-se da Agrimensura, incluem a escrita para jornais e periódicos científicos, contribuição em dicionários escolares, tradução de trabalhos científicos, palestras e escrita de livros teóricos. Infelizmente, todas essas atividades não foram suficientes para afastar Charles e sua esposa da pobreza abjeta. Nos seus últimos anos de vida, ele foi sustentado por um fundo criado por seus admiradores e administrado pelo filósofo William James, seu grande amigo. Embora Peirce tenha publicado muitas obras em diversas áreas, ele deixou mais de 100.000 manuscritos sem publicar. Por causa da dificuldade de estudar suas obras manuscritas, pesquisadores começaram a entender apenas recentemente algumas de suas várias contribuições. Um grupo de pessoas dedica-se a tornar seu trabalho disponível na Internet para trazer melhor apreciação do trabalho de Peirce para o mundo.

EXEMPLO 6 Considere a afirmação

if $x > 0$ **then** $x := x + 1$.

Quando essa declaração é encontrada em um programa, o valor da variável x naquele ponto de execução é inserido em $P(x)$, que é "$x > 0$". Se $P(x)$ é verdadeira para esse valor de x, o comando $x := x + 1$ é executado, logo o valor de x é incrementado em uma unidade. Se $P(x)$ é falsa para esse valor de x, o comando não é executado, e, portanto, o valor de x não é alterado. ◀

Predicados são também usados em programas de computador para verificar se eles sempre produzem uma saída desejada quando é dada uma entrada válida. As declarações que descrevem entradas válidas são conhecidas por **condições iniciais** ou **precondições** e as condições que verificam se as saídas são satisfatórias quando o programa roda são chamadas de **condições finais** ou **pós-condições**. Como ilustra o Exemplo 7, usamos predicados para descrever ambas as condições: precondições e pós-condições. Vamos estudar esse processo com mais detalhes na Seção 4.4.

EXEMPLO 7 Considere o seguinte programa, feito para trocar os valores das variáveis x e y.

```
temp := x
x := y
y := temp
```

Encontre predicados que podem ser usados como precondições e pós-condições para verificar se esse programa é correto. Explique como podemos usá-los para verificar se para toda entrada válida o programa faz o que se pretende.

Solução: Como precondição, precisamos saber se x e y têm certos valores antes de rodar o programa. Então, para essa precondição, podemos usar o predicado $P(x, y)$, no qual $P(x, y)$ é a afirmação "$x = a$ e $y = b$", em que a e b são os valores de x e y antes de rodar o programa. Como queremos verificar se o programa está trocando os valores das duas variáveis, como pós-condição podemos usar $Q(x, y)$, em que $Q(x, y)$ é "$x = b$ e $y = a$".

Para verificar se esse programa sempre faz o que se deseja que faça, suponha que a precondição $P(x, y)$ é satisfeita. Ou seja, supomos que "$x = a$ e $y = b$" é verdadeira. Isso significa que $x = a$ e $y = b$. O primeiro passo do programa, $temp := x$, faz a variável $temp$ receber o valor de x, então, depois desse passo, $x = a$, $temp = a$ e $y = b$. Depois do segundo passo, $x := y$, sabemos que $x = b$, $temp = a$ e $y = b$. Finalmente, depois do terceiro passo, sabemos que $x = b$, $temp = a$ e $y = a$. Conseqüentemente, depois de rodar o programa, a pós-condição $Q(x, y)$ é satisfeita, isto é, "$x = b$ e $y = a$" é verdadeira. ◀

Quantificadores

Quando impomos às variáveis de uma função proposicional algum valor, a declaração resultante torna-se uma proposição e tem um valor-verdade. No entanto, existe uma outra maneira importante, chamada de **quantificação**, para criar proposições a partir de funções proposicionais. A quantificação é um meio de dizer que um predicado é verdadeiro para um conjunto de elementos. Em português, as palavras *muitos, todos, alguns, nenhum* e *poucos* são usadas em quantificações. Vamos nos concentrar em dois tipos de quantificação aqui: a universal, a qual significa que um predicado é verdadeiro para todos os elementos em consideração, e a existencial, a qual nos diz que existe um ou mais elementos para os quais o predicado é verdadeiro. A área da lógica que estuda predicados e quantificadores é chamada de **cálculo de predicados.**

Auto-avaliação

Auto-avaliação

O QUANTIFICADOR UNIVERSAL Muitas afirmações matemáticas referem-se a alguma propriedade que é verdadeira para todos os valores de uma variável em determinado domínio, chamado de **domínio de discurso** (ou de **universo de discurso**), freqüentemente apenas chamado de **domínio**. Essas afirmações são expressas usando quantificação universal. A quantificação universal de $P(x)$ para determinado domínio é a proposição que afirma que $P(x)$ é verdadeira para todos os valores de x pertencentes a esse domínio. Note que o domínio especifica os possíveis valores da variável x. O significado da quantificação universal de $P(x)$ muda quando mudamos o domínio. O domínio deve ser sempre especificado quando usamos um quantificador universal; sem ele, a quantificação universal não está definida.

DEFINIÇÃO 1

A *quantificação universal* de $P(x)$ é a afirmação

"$P(x)$ é válida para todos os valores de x do domínio."

A notação $\forall x P(x)$ indica a quantificação universal de $P(x)$. Aqui $\forall$ é chamado de **quantificador universal**. Lemos $\forall x P(x)$ como "para todo $x P(x)$". Um elemento para o qual $P(x)$ é falsa é chamado de **contra-exemplo** para $\forall x P(x)$.

O significado do quantificador universal é resumido na primeira linha da Tabela 1. Vamos ilustrar o uso do quantificador universal nos exemplos 8–13.

EXEMPLO 8 Seja $P(x)$ a declaração "$x + 1 > x$". Qual é o valor-verdade da quantificação $\forall x P(x)$, no domínio de todos os números reais?

Exemplos Extras

Solução: Como $P(x)$ é verdadeira para todo número real x, a quantificação

$$\forall x P(x)$$

é verdadeira. ◀

Lembre-se: Em geral, é assumido implicitamente que todos os domínios dos quantificadores são não vazios. Note que, se o domínio é vazio, então $\forall x P(x)$ é verdadeira para toda proposição $P(x)$, uma vez que não há elemento no domínio para o qual $P(x)$ é falsa.

Além disso, a quantificação universal, "para todo"[1] pode ser expressa de muitas outras maneiras, incluindo "todos os", "para cada", "dado qualquer", "arbitrariamente", "para cada" e "para qualquer".

TABELA 1 Quantificadores.

Sentença	*Quando é verdadeira?*	*Quando é falsa?*
$\forall x P(x)$	$P(x)$ é verdadeira para todo x.	Existe um x tal que $P(x)$ é falsa.
$\exists x P(x)$	Existe um x tal que $P(x)$ é verdadeira.	$P(x)$ é falsa para todo x.

[1] N.T.: Neste ponto, o livro original faz menção aos termos equivalentes em inglês que podem causar ambigüidade. Essas ambigüidades não devem ser consideradas em português.

Uma declaração $\forall x P(x)$ é falsa, em que $P(x)$ é uma função proposicional, se e somente se $P(x)$ não é sempre verdadeira para os valores de x no domínio. Uma maneira de mostrar que $P(x)$ não é sempre verdadeira no domínio é achar um contra-exemplo para a declaração $\forall x P(x)$. Note que um único contra-exemplo é tudo de que precisamos para estabelecer que $\forall x P(x)$ é falsa. O Exemplo 9 ilustra como contra-exemplos são usados.

EXEMPLO 9 Seja $Q(x)$ a declaração "$x < 2$". Qual o valor-verdade da quantificação $\forall x Q(x)$, em que o domínio consiste em todos os números reais?

Solução: $Q(x)$ não é verdadeira para todo número real x, porque, por exemplo, $Q(3)$ é falsa. Isto é, $x = 3$ é um contra-exemplo para a declaração $\forall x Q(x)$. Logo

$$\forall x Q(x)$$

é falsa. ◀

EXEMPLO 10 Suponha que $P(x)$ seja "$x^2 > 0$". Para mostrar que $\forall x P(x)$ é falsa onde o universo de discurso consiste em todos os números inteiros, damos um contra-exemplo. Vemos que $x = 0$ é um contra-exemplo, pois $x^2 = 0$ quando $x = 0$, então x^2 não é maior que 0 quando $x = 0$. ◀

Procurar por contra-exemplos em proposições universalmente quantificadas é uma importante atividade no estudo da matemática, como veremos nas seções seguintes deste livro.

Quando todos os elementos do domínio podem ser listados — seja $x_1, x_2, \ldots, x_n$ —, segue-se que a quantificação universal $\forall x P(x)$ é o mesmo que a conjunção

$$P(x_1) \wedge P(x_2) \wedge \ldots \wedge P(x_n),$$

pois esta conjunção é verdadeira se e somente se $P(x_1), P(x_2), \ldots, P(x_n)$ forem todas verdadeiras.

EXEMPLO 11 Qual o valor-verdade de $\forall x P(x)$, em que $P(x)$ é a proposição "$x^2 < 10$" e o domínio é o conjunto dos inteiros positivos que não excedem 4?

Solução: A declaração $\forall x P(x)$ é o mesmo que a conjunção

$$P(1) \wedge P(2) \wedge P(3) \wedge P(4),$$

pois o domínio é formado por esses quatro elementos. Como $P(4)$, que é a expressão "$4^2 < 10$", é falsa, segue-se que $\forall x P(x)$ é falsa. ◀

EXEMPLO 12 O que significa dizer $\forall x N(x)$ se $N(x)$ é "O computador x está conectado à rede" e o domínio são todos os computadores do campus?

Solução: A declaração $\forall x N(x)$ significa que, para todo computador x no campus, x está conectado à rede. Em português, a declaração pode ser expressa por "Todo computador no campus está conectado à rede". ◀

Apontamos anteriormente o fato de que a especificação do domínio é primordial e obrigatória quando quantificadores são usados. O valor-verdade da proposição quantificada freqüentemente depende do domínio, como mostra o Exemplo 13.

EXEMPLO 13 Qual o valor-verdade de $\forall x(x^2 \geq x)$ se o domínio consiste em todos os números reais? E qual o valor-verdade dessa proposição se o domínio são todos os números inteiros?

Solução: A quantificação universal $\forall x(x^2 \geq x)$, com domínio nos números reais, é falsa. Por exemplo, $(\frac{1}{2})^2 \not\geq \frac{1}{2}$. Note que $x^2 \geq x$ se e somente se $x^2 - x = x(x-1) \geq 0$. Conseqüentemente, $x^2 \geq x$ se e somente se $x \leq 0$ ou $x \geq 1$. Daqui segue que $\forall x(x^2 \geq x)$ é falsa se o domínio consiste em todos os números reais (pois a inequação não é válida para os números reais entre 0 e 1). No entanto, se o domínio são os números inteiros, $\forall x(x^2 \geq x)$ é verdadeira, pois não há números inteiros entre 0 e 1. ◀

QUANTIFICADOR EXISTENCIAL Muitas proposições matemáticas dizem que existe um elemento com determinada propriedade. Essas proposições são expressas usando a quantificação existencial. Com a quantificação existencial, construímos uma proposição que é verdadeira se e somente se $P(x)$ é verdadeira para, pelo menos, um valor no domínio.

DEFINIÇÃO 2

A *quantificação existencial* de $P(x)$ é a proposição

"Existe um elemento x no domínio tal que $P(x)$."

Usamos a notação $\exists x P(x)$ para a quantificação existencial de $P(x)$. Aqui $\exists$ é chamado de **quantificador existencial**.

Um domínio deve sempre ser especificado quando uma proposição $\exists x P(x)$ é usada. Até mesmo porque seu significado muda quando mudamos o domínio. Sem a especificação de um domínio, a expressão $\exists x P(x)$ não tem sentido. A quantificação existencial $\exists x P(x)$ é lida como

"Existe um x tal que $P(x)$."
"Existe pelo menos um x tal que $P(x)$."

ou

"Para algum x $P(x)$."

No lugar da palavra "existe", podemos também expressar a quantificação existencial de muitas outras maneiras, tais como usar as palavras "para algum", "para pelo menos um" ou " há".

O significado do quantificador existencial é resumido na segunda linha da Tabela 1. Vamos ilustrar o uso do quantificador existencial nos exemplos 14–16.

EXEMPLO 14 Seja $P(x)$ a expressão "$x > 3$". Qual o valor-verdade da quantificação $\exists x P(x)$ no domínio dos números reais?

Exemplos Extras

Solução: Como "$x > 3$" é verdadeira para alguns números reais — por exemplo, quando $x = 4$ —, a quantificação existencial de $P(x)$, que é $\exists x P(x)$, é verdadeira. ◀

Observe que a proposição $\exists x P(x)$ é falsa se e somente se não existe elemento x no domínio para o qual $P(x)$ é verdadeira. Ou seja, $\exists x P(x)$ é falsa se e somente se $P(x)$ é falsa para todo elemento do domínio. Vamos ilustrar essa observação no Exemplo 15.

EXEMPLO 15 Seja $Q(x)$ a expressão "$x = x + 1$". Qual o valor-verdade da quantificação $\exists x Q(x)$ no domínio dos números reais?

Solução: Como $Q(x)$ é falsa para todos os números reais, a quantificação existencial de $Q(x)$, que é $\exists x Q(x)$, é falsa. ◀

Lembre-se: Em geral, é assumido implicitamente que todos os domínios dos quantificadores são não vazios. Note que se o domínio é vazio, então $\exists x Q(x)$ é falsa para toda função proposicional $Q(x)$, uma vez que não há elemento no domínio que valide $Q(x)$.

Quando todos os elementos do domínio podem ser listados — seja $x_1, x_2, \ldots, x_n$ —, a quantificação existencial $\exists x P(x)$ é a mesma que a disjunção

$$P(x_1) \vee P(x_2) \vee \cdots \vee P(x_n),$$

pois essa disjunção é verdadeira se e somente se pelo menos uma das $P(x_1), P(x_2), \ldots, P(x_n)$ for verdadeira.

EXEMPLO 16 Qual o valor-verdade de $\exists x\, P(x)$, em que $P(x)$ é a proposição "$x^2 > 10$" e o domínio é o conjunto dos inteiros positivos que não excedem 4?

Solução: Como o domínio é $\{1, 2, 3, 4\}$, a proposição $\exists x P(x)$ é a mesma que a disjunção

$$P(1) \vee P(2) \vee P(3) \vee P(4).$$

Como $P(4)$, que é a proposição "$4^2 > 10$", é verdadeira, segue-se que $\exists x\, P(x)$ é verdadeira. ◀

Às vezes é interessante dar uma passada por todos os termos do domínio ou fazer uma procura entre esses termos quando estamos determinando valores-verdade de uma quantificação. Suponha que temos n objetos no domínio para uma variável x. Para determinar quando $\forall x P(x)$ é verdadeira, podemos dar uma passada por todos os valores de x para ver se $P(x)$ é sempre verdadeira. Se encontrarmos um valor de x para o qual $P(x)$ é falsa, então, teremos mostrado que $\forall x P(x)$ é falsa. Caso contrário, $\forall x P(x)$ será verdadeira. Para ver quando $\exists x P(x)$ é verdadeira, damos uma passada pelos n valores de x procurando um valor para o qual $P(x)$ é verdadeira. Se nunca encontrarmos um tal valor de x, teremos, então, determinado que $\exists x P(x)$ é falsa. (Note que esse procedimento de procura não se aplica quando existem infinitos valores de x no domínio. No entanto, é uma maneira possível de pensar sobre os valores-verdade das quantificações.)

Outras Quantificações

Agora temos introduzido os quantificadores universal e existencial. Esses são os mais importantes quantificadores em matemática e em ciência da computação. No entanto, existe um número não limitado de quantificadores que podemos definir, tais como "existem exatamente dois", "existem não mais de três", "existem pelo menos 100", e assim por diante. Desses outros quantificadores, um dos mais freqüentemente vistos é o **quantificador de unicidade**, indicado por $\exists!$ ou $\exists_1$. A notação $\exists! x P(x)$ [ou $\exists_1 x P(x)$] indica que "Existe um único x tal que $P(x)$ é verdadeira". Outras frases podem ser usadas para a quantificação de unicidade, incluindo "existe exatamente um" e "existe um e somente um". Observe que podemos usar quantificadores e lógica proposicional para expressar unicidade (veja o Exercício 52 na Seção 1.4), então podemos nos esquivar do quantificador de unicidade. Geralmente, é melhor trabalhar com os quantificadores universal e existencial, pois as regras de inferência para esses quantificadores podem ser usadas.

Quantificadores com Domínio Restrito

Uma notação abreviada é freqüentemente usada para restringir o domínio de um quantificador. Nessa notação, uma condição que a variável deve satisfazer é incluída depois do quantificador. Esse fato é ilustrado no Exemplo 17. Vamos também descrever outras formas de notação que envolvem elementos de conjuntos na Seção 2.1.

EXEMPLO 17 O que as proposições $\forall x < 0\ (x^2 > 0)$, $\forall y \neq 0\ (y^3 \neq 0)$ e $\exists z > 0\ (z^2 = 2)$ significam, em que o domínio em cada um dos casos é o conjunto dos números reais?

Solução: A proposição $\forall x < 0\ (x^2 > 0)$ fala sobre qualquer número real x com $x < 0$, $x^2 > 0$. Ou seja, ela diz que "O quadrado de todo número negativo é positivo". A proposição é o mesmo que $\forall x(x < 0 \rightarrow x^2 > 0)$.

A proposição $\forall y \neq 0\ (y^3 \neq 0)$ fala que, para qualquer número real y com $y \neq 0$, teremos $y^3 \neq 0$. Ou seja, ela diz que "O cubo de um número não nulo é também não nulo". Note que esta proposição é equivalente a $\forall y(y \neq 0 \rightarrow y^3 \neq 0)$.

Finalmente, a proposição $\exists z > 0\ (z^2 = 2)$ fala que existe um número real z com $z > 0$, tal que $z^2 = 2$. Ou seja, ela diz que "Existe um número real positivo tal que seu quadrado é igual a 2". Essa proposição é equivalente a $\exists z(z > 0 \wedge z^2 = 2)$. ◀

Note que a restrição de um quantificador universal é a mesma que o quantificador universal de uma proposição condicional. Por exemplo, $\forall x < 0\ (x^2 > 0)$ é uma outra maneira de expressar $\forall x(x < 0 \rightarrow x^2 > 0)$. Por outro lado, a restrição de um quantificador existencial é a mesma que o quantificador existencial de uma conjunção. Por exemplo, $\exists z > 0\ (z^2 = 2)$ é uma outra maneira de expressar $\exists z(z > 0 \wedge z^2 = 2)$.

Prioridade dos Quantificadores

Os quantificadores $\forall$ e $\exists$ têm prioridade maior que todos os operadores lógicos do cálculo proposicional. Por exemplo, $\forall x P(x) \vee Q(x)$ é a disjunção de $\forall x P(x)$ e $Q(x)$. Em outras palavras, ela significa $(\forall x P(x)) \vee Q(x)$ em vez de $\forall x(P(x) \vee Q(x))$.

Ligando Variáveis

Quando um quantificador é usado na variável x, dizemos que essa ocorrência da variável é **ligada**. Uma ocorrência de uma variável que não é ligada por um quantificador ou não representa um conjunto de valores particulares é chamada de **variável livre**. Todas as variáveis que ocorrem em uma função proposicional devem ser ligadas ou devem representar um conjunto de valores particulares para ser uma proposição. Isso pode ser feito usando uma combinação de quantificadores universais, existenciais ou dando algum valor para as variáveis.

A parte da expressão lógica à qual um quantificador é aplicado é chamada de **escopo** do quantificador. Conseqüentemente, uma variável é livre se ela não está sob o escopo de algum quantificador na fórmula em que aparece essa variável.

EXEMPLO 18 Na afirmação $\exists x(x + y = 1)$, a variável x é ligada pelo quantificador existencial, mas a variável y é livre, pois não é ligada a nenhum quantificador, nem assume nenhum valor específico. Isso ilustra que, na declaração $\exists x(x + y = 1)$, x é ligada e y é livre.

Na afirmação $\exists x(P(x) \wedge Q(x)) \vee \forall x R(x)$, todas as variáveis são ligadas. O escopo do primeiro quantificador é $\exists x$, pois $P(x) \wedge Q(x)$ é aplicado apenas a $\exists x$, e não ao resto da expressão. Similarmente, o escopo do segundo quantificador, $\forall x$, nesta expressão é $R(x)$. Isto é, o quantificador existencial atua sobre a variável x em $P(x) \wedge Q(x)$ e o quantificador universal $\forall x$ atua sobre a variável x em $R(x)$. Observe que poderíamos ter escrito nossa afirmação usando duas variáveis diferentes x

e y, como $\exists x(P(x) \wedge Q(x)) \vee \forall y R(y)$, pois o escopo dos dois quantificadores não se sobrepõe. O leitor deve estar ciente de que, no uso comum, a mesma letra é freqüentemente usada para representar variáveis ligadas por diferentes quantificadores com escopo que não se sobrepõe. ◀

Equivalências Lógicas que Envolvem Quantificadores

Na Seção 1.2, introduzimos a noção de equivalências lógicas de proposições compostas. Podemos estender essa noção a expressões que envolvem predicados e quantificadores.

DEFINIÇÃO 3 Sentenças que envolvem predicados e quantificadores são *logicamente equivalentes* se e somente se elas têm o mesmo valor-verdade quaisquer que sejam os predicados substituídos nessas sentenças e qualquer que seja o domínio de discurso para as variáveis nessas funções proposicionais. Usamos a notação $S \equiv T$ para indicar que as duas declarações que envolvem predicados e quantificadores são logicamente equivalentes.

O Exemplo 19 ilustra como duas declarações que envolvem predicados e quantificadores são logicamente equivalentes.

EXEMPLO 19 Mostre que $\forall x(P(x) \wedge Q(x))$ e $\forall x P(x) \wedge \forall x Q(x)$ são logicamente equivalentes (em que o mesmo domínio é usado nas duas). Essa equivalência lógica mostra que podemos distribuir o quantificador universal sobre a conjunção. Além disso, podemos distribuir o quantificador existencial sobre a disjunção. No entanto, não podemos distribuir o universal sobre a disjunção nem o existencial sobre a conjunção. (Veja os exercícios 50 e 51.)

Solução: Para mostrar que essas sentenças são logicamente equivalentes, devemos mostrar que elas têm sempre o mesmo valor-verdade, não importando o que são os predicados P e Q, e não importando qual seja o domínio usado. Suponha que tenhamos predicados particulares P e Q, com um domínio comum. Podemos mostrar que $\forall x(P(x) \wedge Q(x))$ e $\forall x P(x) \wedge \forall x Q(x)$ são logicamente equivalentes fazendo duas coisas. Primeiro, mostramos que se $\forall x(P(x) \wedge Q(x))$ é verdadeira, então $\forall x P(x) \wedge \forall x Q(x)$ é verdadeira. Depois, mostramos que, se $\forall x P(x) \wedge \forall x Q(x)$ é verdadeira, então $\forall x(P(x) \wedge Q(x))$ é verdadeira.

Então, suponha que $\forall x(P(x) \wedge Q(x))$ seja verdadeira. Isso significa que se a está no domínio, então $P(a) \wedge Q(a)$ é verdadeira. Logo, $P(a)$ é verdadeira e $Q(a)$ é verdadeira. Como $P(a)$ é verdadeira e $Q(a)$ é verdadeira para todo elemento do domínio, podemos concluir que $\forall x P(x)$ e $\forall x Q(x)$ são ambas verdadeiras. Isso significa que $\forall x P(x) \wedge \forall x Q(x)$ é verdadeira.

Agora podemos supor que $\forall x P(x) \wedge \forall x Q(x)$ é verdadeira. Disso segue que $\forall x P(x)$ e $\forall x Q(x)$ são ambas verdadeiras. Logo, se a está no domínio, então $P(a)$ é verdadeira e $Q(a)$ é verdadeira (como $P(x)$ e $Q(x)$ são verdadeiras para todos os elementos do domínio, não há nenhum problema em usar o mesmo valor a). Disso segue que, para todo a do domínio, $P(a) \wedge Q(a)$ é verdadeira. Portanto, $\forall x(P(x) \wedge Q(x))$ é verdadeira. Podemos, então, concluir que

$$\forall x(P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x).$$ ◀

Negando Expressões Quantificadas

Freqüentemente vamos querer considerar a negação das expressões quantificadas. Por exemplo, considere a negação da expressão

"Todo estudante na sua classe teve aulas de cálculo."

Essa expressão é uma quantificação universal, nominalmente,

$$\forall x P(x),$$

Auto-avaliação

em que $P(x)$ é a declaração "x teve aulas de cálculo" e o domínio consiste em todos os estudantes de sua classe. A negação dessa proposição é "Não é o caso de todos os alunos de sua classe terem feito aulas de cálculo". Isso é equivalente a "Existe um estudante em sua classe que não teve aula de cálculo". E isso é simplesmente a quantificação existencial da negação da função proposicional original, nominalmente,

$$\exists x \neg P(x).$$

Esse exemplo ilustra a seguinte equivalência lógica:

$$\neg \forall x P(x) \equiv \exists x \neg P(x).$$

Para mostrar que $\neg \forall x P(x)$ e $\exists x(x)$ são logicamente equivalentes, não importando o que significa a função proposicional $P(x)$ e tampouco qual o domínio, primeiro note que $\neg \forall x P(x)$ é verdadeira se e somente se $\forall x P(x)$ é falsa. Depois, note que $\forall x P(x)$ é falsa se e somente se existe um elemento x no domínio para o qual $\neg P(x)$ é verdadeira. Finalmente, observe que existe um elemento x no domínio, tal que $\neg P(x)$ é verdadeira se e somente se $\exists x \neg P(x)$ é verdadeira. Colocando esses passos em seqüência, podemos concluir que $\neg \forall x P(x)$ e $\exists x \neg P(x)$ são logicamente equivalentes.

Suponha que queiramos negar uma quantificação existencial. Por exemplo, considere a expressão "Existe um estudante na sua classe que teve aulas de cálculo". Este é o quantificador existencial

$$\exists x Q(x),$$

em que $Q(x)$ é a declaração "x teve aulas de cálculo". A negação dessa frase é a proposição "Não é o caso de existir um estudante na sua classe que teve aulas de cálculo". Que é equivalente a "Todo estudante nesta classe não teve aulas de cálculo", que é a quantificação universal da negação da função proposicional original, ou, escrito em linguagem dos quantificadores,

$$\forall x \neg Q(x).$$

Esse exemplo ilustra a equivalência

$$\neg \exists x Q(x) \equiv \forall x \neg Q(x).$$

Para mostrar que $\neg \exists x Q(x) \equiv \forall x \neg Q(x)$ são logicamente equivalentes, não importando o que significa a função proposicional $Q(x)$ e tampouco qual o domínio, primeiro note que $\neg \exists x Q(x)$ é verdadeira se e somente se $\exists x Q(x)$ é falsa. E isso é verdadeiro se e somente se não existe um elemento x no domínio para o qual $Q(x)$ é verdadeira. Depois, note que não existe x no domínio para o qual $Q(x)$ é verdadeira se e somente se $Q(x)$ é falsa para todo x no domínio. Finalmente, observe que $Q(x)$ é falsa para todo x no domínio se e somente se $\neg Q(x)$ é verdadeira para todo x no domínio, que só pode ocorrer se e somente se $\forall x \neg Q(x)$ é verdadeira. Colocando esses passos em seqüência, vemos que $\neg \exists x Q(x)$ é verdadeira se e somente se $\forall x \neg Q(x)$ é verdadeira. E concluímos que eles são logicamente equivalentes.

As regras para negações de quantificadores são chamadas de **leis de De Morgan para quantificadores**. Essas regras estão resumidas na Tabela 2.

TABELA 2 Leis de De Morgan para Quantificadores.			
Negação	*Sentença Equivalente*	*Quando a Negação é Verdadeira?*	*Quando é Falsa?*
$\neg\exists x P(x)$	$\forall x \neg P(x)$	Para todo x, $P(x)$ é falsa.	Existe um x para o qual $P(x)$ é verdadeira.
$\neg\forall x P(x)$	$\exists x \neg P(x)$	Existe um x para o qual $P(x)$ é falsa.	Para todo x, $P(x)$ é verdadeira.

Lembre-se: Quando o domínio de um predicado $P(x)$ consiste em n elementos, em que n é um número inteiro positivo, as regras de negação para proposições quantificadas são exatamente como as leis de De Morgan discutidas na Seção 1.2. Por isso, chamamos essas leis de leis de De Morgan para quantificadores. Quando o domínio tem n elementos $x_1, x_2, \ldots, x_n$, segue que $\neg\forall x P(x)$ é o mesmo que $\neg(P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n))$, que é equivalente a $\neg P(x_1) \vee \neg P(x_2) \vee \cdots \vee \neg P(x_n)$ pela lei de De Morgan, e isso é o mesmo que $\exists x \neg P(x)$. De maneira análoga, $\neg\exists x P(x)$ é o mesmo que $\neg(P(x_1) \vee P(x_2) \vee \cdots \vee P(x_n))$, que pela lei de De Morgan é equivalente a $\neg P(x_1) \wedge \neg P(x_2) \wedge \cdots \wedge \neg P(x_n)$ e isso é o mesmo que $\forall x \neg P(x)$.

Ilustramos a negação de proposições quantificadas nos exemplos 20 e 21.

EXEMPLO 20 Quais as negações de "Existe um político honesto" e "Todos os brasileiros comem churrasco"?

Solução: Seja $H(x)$ correspondente a "x é honesto". Então, a proposição "Existe um político honesto" é representada por $\exists x H(x)$, em que o domínio consiste em todos os políticos. A negação dessa declaração é $\neg\exists x H(x)$, que é equivalente a $\forall x \neg H(x)$. Essa negação pode ser expressa por "Todos os políticos são desonestos" (ou "Todos os políticos são não honestos", mas, dependendo da língua em que se fala, essa última pode gerar uma ambigüidade; logo, preferimos a primeira).

Exemplos Extras

Seja $C(x)$ correspondente a "x come churrasco". Então, a proposição "Todos os brasileiros comem churrasco" é representada por $\forall x C(x)$, em que o domínio consiste em todos os brasileiros. A negação dessa proposição é representada por $\neg\forall x C(x)$, que é equivalente a $\exists x \neg C(x)$. Essa negação pode ser expressa de muitas maneiras diferentes, incluindo "Alguns brasileiros não comem churrasco" ou "Existe um brasileiro que não come churrasco". ◀

EXEMPLO 21 Quais as negações das proposições $\forall x(x^2 > x)$ e $\exists x(x^2 = 2)$?

Solução: A negação de $\forall x(x^2 > x)$ é a proposição $\neg\forall x(x^2 > x)$, que é equivalente a $\exists x \neg(x^2 > x)$. Esta pode ser reescrita como $\exists x(x^2 \leq x)$. A negação de $\exists x(x^2 = 2)$ é a proposição $\neg\exists x(x^2 = 2)$, que é equivalente a $\forall x \neg(x^2 = 2)$. Essa pode ser reescrita como $\forall x(x^2 \neq 2)$. Os valores-verdade dessas proposições dependem do domínio. ◀

Usaremos as leis de De Morgan para quantificadores no Exemplo 22.

EXEMPLO 22 Mostre que $\neg\forall x(P(x) \rightarrow Q(x))$ e $\exists x(P(x) \wedge \neg Q(x))$ são logicamente equivalentes.

Solução: Pela lei de De Morgan para quantificadores universais, sabemos que $\neg\forall x(P(x) \rightarrow Q(x))$ e $\exists x(\neg(P(x) \rightarrow Q(x)))$ são logicamente equivalentes. Pela quinta equivalência lógica da Tabela 7 da Seção 1.2, sabemos que $\neg(P(x) \rightarrow Q(x))$ e $P(x) \wedge \neg Q(x)$ são logicamente equivalentes para todo x. Como podemos substituir uma expressão logicamente equivalente por outra, em uma equivalência lógica, segue que $\neg\forall x(P(x) \rightarrow Q(x))$ e $\exists x(P(x) \wedge \neg Q(x))$ são logicamente equivalentes. ◀

Traduzindo do Português para Expressões Lógicas

Traduzir sentenças em português (ou outra linguagem natural) para expressões lógicas é uma tarefa crucial em matemática, lógica de programação, inteligência artificial, engenharia de software e muitas outras disciplinas. Começamos a estudar esse tópico na Seção 1.1, onde usamos proposições para expressar sentenças em expressões lógicas. Naquela discussão, nós propositalmente evitamos sentenças em que suas traduções necessitavam de predicados e quantificadores. Traduzir do português para expressões lógicas torna-se mais complicado quando quantificadores são necessários. Além do mais, podem existir muitas maneiras de traduzir uma sentença particular. (Portanto, não existe uma receita que pode ser seguida passo a passo.) Vamos usar alguns exemplos para ilustrar como traduzir do português para expressões lógicas. O objetivo nessa tradução é produzir expressões lógicas simples e usuais. Nesta seção, restringiremos a sentenças que podem ser traduzidas usando apenas um quantificador; na próxima seção, veremos sentenças mais complicadas que requerem muitos quantificadores.

EXEMPLO 23 Expresse a sentença "Todo estudante desta classe estudou cálculo", usando predicados e quantificadores.

Solução: Primeiro, reescrevemos a sentença para identificar claramente qual o quantificador apropriado para usar. Fazendo isso, obtemos:

"Para cada estudante desta classe, este estudante estudou cálculo."

Exemplos Extras

Depois, introduzimos uma variável e a sentença torna-se

"Para cada estudante x desta classe, x estudou cálculo."

Continuando, introduzimos $C(x)$, que é o predicado "x estudou cálculo". Conseqüentemente, se o domínio para x consiste nos estudantes desta classe, podemos traduzir nossa sentença para $\forall x\, C(x)$.

No entanto, existem outras maneiras corretas; diferentes domínios e predicados que podem ser usados. A tradução que escolhemos depende do raciocínio subseqüente que queremos levar a cabo. Por exemplo, podemos estar interessados em um grupo de pessoas mais amplo que aquele dessa classe. Se mudarmos o domínio para todas as pessoas, precisaremos expressar nossa sentença por

"Para cada pessoa x, se x é um estudante desta classe, então x estudou cálculo."

Se $S(x)$ representa a sentença "x é um estudante desta classe", vemos que nossa sentença pode ser expressa por $\forall x(S(x) \rightarrow C(x))$. [*Cuidado!* Nossa sentença *não* pode ser expressa por $\forall x(S(x) \wedge C(x))$, pois essa expressão diz que todas as pessoas são estudantes desta classe e já estudaram cálculo!]

Finalmente, quando estamos interessados em relacionar as pessoas com a matéria cálculo, podemos preferir um predicado com duas variáveis $Q(x, y)$ para a sentença "estudante x estudou a matéria y". E podemos substituir $C(x)$ por $Q(x,$ cálculo) em ambas as traduções anteriores e obter $\forall x\, Q(x,$ cálculo) ou $\forall x(S(x) \rightarrow Q(x,$ cálculo)). ◀

No Exemplo 23, mostramos diferentes modos de expressar a mesma sentença usando predicados e quantificadores. No entanto, devemos sempre adotar o mais simples, que é adequado para usar em nosso raciocínio subseqüente.

EXEMPLO 24 Expresse as sentenças "Algum estudante da classe visitou o México" e "Todo estudante da classe visitou Canadá ou México" usando predicados e quantificadores.

Solução: A sentença "Algum estudante da classe visitou o México" significa que

"Existe um estudante da classe com a propriedade de que o estudante visitou o México."

Podemos introduzir uma variável x, e a sentença passará a

"Existe um estudante x da classe com a propriedade de que x visitou o México."

Introduzimos $M(x)$, que é a sentença "x visitou o México". Se o domínio consiste em todos os estudantes da classe, podemos traduzir essa primeira sentença por $\exists x M(x)$.

No entanto, se estamos interessados nas pessoas além das pessoas da classe, podemos olhar para a sentença de maneira um pouco diferente. Nossa sentença pode ser expressa por

"Existe uma pessoa x que tem as propriedades de x ser estudante da classe e x visitou o México."

Nesse caso, o domínio da variável x consiste em todas as pessoas. Introduzimos $S(x)$ para representar "x é estudante da classe". Nossa solução fica $\exists x(S(x) \wedge M(x))$ porque a sentença diz que existe uma pessoa x que é estudante e visitou o México. [*Cuidado!* Nossa sentença *não* pode ser expressa por $\exists x(S(x) \rightarrow M(x))$, que é verdadeira para qualquer pessoa que não esteja na classe, pois, nesse caso, para uma pessoa x, $S(x) \rightarrow M(x)$ torna-se $\mathbf{F} \rightarrow \mathbf{V}$ ou $\mathbf{F} \rightarrow \mathbf{F}$, ambas verdadeiras.]

Similarmente, a segunda sentença pode ser expressa por

"Para cada estudante da classe x, x tem a propriedade de x ter visitado o México ou x ter visitado o Canadá."

(Note que estamos assumindo o *ou* inclusivo, em vez do *ou* exclusivo.) Seja $C(x)$ a sentença "x visitou o Canadá". Seguindo o nosso raciocínio, vemos que, com o domínio consistindo nos alunos da classe, a segunda sentença pode ser expressa por $\forall x(C(x) \vee M(x))$. No entanto, se o domínio consiste em todas as pessoas, nossa sentença pode ser expressa por

"Para cada pessoa x, se x é estudante da classe, então x tem a propriedade de x ter visitado o México ou x ter visitado o Canadá."

Nesse caso, a sentença pode ser expressa por $\forall x(S(x) \rightarrow (C(x) \vee M(x)))$.

Em vez de usar $M(x)$ e $C(x)$ para representar que x visitou o México e x visitou o Canadá, respectivamente, podemos usar um predicado binário $V(x, y)$ para representar "x visitou o país y". Nesse caso, $V(x, \text{México})$ e $V(x, \text{Canadá})$ terão o mesmo significado de $M(x)$ e $C(x)$ e podem substituí-los em nossas respostas. Se estivermos trabalhando com muitas sentenças que envolvam pessoas que visitam diferentes países, podemos preferir trabalhar com esse predicado com duas variáveis. Caso contrário, por simplicidade, devemos escolher predicados com uma variável. ◀

Usando Quantificadores em Sistemas de Especificações

Na Seção 1.1, usamos proposições para representar sistemas de especificações. No entanto, muitos sistemas de especificações envolvem predicados e quantificações. Isso é ilustrado no Exemplo 25.

EXEMPLO 25 Use predicados e quantificadores para expressar o sistema de especificações "Todo e-mail com tamanho maior que um megabyte será comprimido" e "Se um usuário estiver ativo, ao menos um link de rede estará habilitado".

Exemplos Extras

Solução: Seja $S(m, y)$ o predicado "E-mail m tem tamanho maior que y megabytes", em que o domínio de m consiste em todas as mensagens de e-mail e y é um número real positivo, e seja $C(m)$ o predicado "O e-mail m será comprimido". Então, a especificação "Todo e-mail com

tamanho maior que um megabyte será comprimido" pode ser representada por $\forall m(S(m,1) \rightarrow C(m))$.

Seja $A(u)$ o predicado "O usuário u está ativo", em que a variável u tem como domínio todos os usuários, e seja $S(n, x)$ o predicado "O link de rede n está no estado x", em que n tem como domínio todos os links de rede e x tem como domínio os estados possíveis de cada link. Então, a especificação "Se um usuário estiver ativo, ao menos um link de rede estará habilitado" pode ser expressa por $\exists u A(u) \rightarrow \exists n S(n,$ habilitado). ◀

Exemplos de Lewis Carroll

Lewis Carroll (pseudônimo de Charles Lutwidge Dodgson), o autor de *Alice no País das Maravilhas*, é também autor de muitos trabalhos sobre lógica simbólica. Seu livro contém muitos exemplos de raciocínio usando quantificadores. Os exemplos 26 e 27 são do livro *Symbolic Logic*; outros exemplos contidos neste livro são dados nos exercícios no final desta seção. Esses exemplos ilustram como quantificadores são usados para expressar vários tipos de sentenças.

EXEMPLO 26 Considere estas sentenças. As duas primeiras são chamadas de *premissas* e a terceira é chamada de *conclusão*. O conjunto inteiro é chamado de *argumento*.

"Todos os leões são selvagens."
"Alguns leões não bebem café."
"Algumas criaturas selvagens não bebem café."

(Na Seção 1.5 vamos discutir a questão de determinar quando a conclusão é uma conseqüência válida das premissas. Neste exemplo, é.) Sejam $P(x)$, $Q(x)$ e $R(x)$ as sentenças "x é um leão", "x é selvagem" e "x bebe café", respectivamente. Assumindo que o domínio consiste em todas as criaturas, expresse as sentenças do argumento usando quantificadores e $P(x)$, $Q(x)$ e $R(x)$.

Solução: Podemos expressar essas sentenças como:

$\forall x(P(x) \rightarrow Q(x))$.
$\exists x(P(x) \wedge \neg R(x))$.
$\exists x(Q(x) \wedge \neg R(x))$.

Note que a segunda sentença não pode ser escrita por $\exists x(P(x) \rightarrow \neg R(x))$. A razão é que $P(x) \rightarrow \neg R(x)$ é verdadeira toda vez que x não é um leão, logo $\exists x(P(x) \rightarrow \neg R(x))$ é verdadeira sempre que existir uma criatura que não seja um leão, mesmo que todo leão beba café. Similarmente, a terceira sentença não pode ser expressa por

$\exists x(Q(x) \rightarrow \neg R(x))$. ◀

Links

CHARLES LUTWIDGE DODGSON (1832–1898) Conhecemos Charles Dodgson como Lewis Carroll — pseudônimo que ele usou em seus escritos sobre lógica. Dodgson, filho de um clérigo, foi o terceiro filho de um total de 11 crianças; de todas, ele era o único que gaguejava. Ele ficava constrangido na presença de adultos e é dito que ele falava sem gaguejar apenas com jovens garotas, muitas com as quais ele se divertia, se correspondia e as fotografava (algumas vezes em poses que hoje seriam consideradas inapropriadas). Embora atraído por jovens garotas, ele era extremamente puritano e religioso. Sua amizade com as três filhas jovens de Dean Liddell inspirou-o a escrever *Alice no País das Maravilhas,* que lhe trouxe dinheiro e fama.

Dodgson formou-se em Oxford em 1854 e obteve seu título de mestre em 1857. Ele foi indicado como professor em matemática na Christ Church College, Oxford, em 1855. Foi ordenado na Igreja da Inglaterra em 1861, mas nunca praticou seu ministério. Seus escritos incluem artigos e livros sobre geometria, determinantes e a matemática de torneios e eleições. (Também usou o pseudônimo Lewis Carroll em muitos de seus trabalhos de lógica recreacional.)

EXEMPLO 27 Considere estas sentenças, das quais as três primeiras são premissas e a quarta é uma conclusão válida.

"Todos os beija-flores são ricamente coloridos."
"Nenhum pássaro grande vive de néctar."
"Pássaros que não vivem de néctar são monótonos nas cores."
"Beija-flores são pequenos."

Sejam $P(x)$, $Q(x)$, $R(x)$ e $S(x)$ as sentenças "x é um beija-flor", "x é grande", "x vive de néctar" e "x é ricamente colorido", respectivamente. Assumindo que o domínio consiste em todos os pássaros, expresse as sentenças do argumento usando quantificadores e $P(x)$, $Q(x)$, $R(x)$ e $S(x)$.

Solução: Podemos expressar as sentenças do argumento por

$\forall x(P(x) \rightarrow S(x))$.
$\neg \exists x(Q(x) \wedge R(x))$.
$\forall x(\neg R(x) \rightarrow \neg S(x))$.
$\forall x(P(x) \rightarrow \neg Q(x))$.

(Note que assumimos que "pequenos" é o mesmo que "não grandes" e que "monótono nas cores" é o mesmo que "não ricamente colorido". Para mostrar que a quarta sentença é uma conclusão válida a partir das três primeiras, precisamos do uso de regras de inferência, que serão discutidas na Seção 1.5.) ◀

Programação Lógica

Um importante tipo de linguagem de programação é designado a raciocinar usando regras de predicados lógicos. Prolog (de *Pro*gramming in *Log*ic), desenvolvido na década de 1970 por cientistas computacionais que trabalhavam na área de inteligência artificial, é um exemplo dessas linguagens. Programas em Prolog incluem um conjunto de declarações que consistem em dois tipos de sentenças, ***Prolog facts*** e ***Prolog rules*** (fatos Prolog e regras Prolog). *Prolog facts* definem predicados que especificam os elementos que satisfazem esses predicados. *Prolog rules* são usados para definir novos predicados que usam aqueles que já estão definidos em *Prolog facts*. O Exemplo 28 ilustra essas noções.

Links

EXEMPLO 28 Considere um programa Prolog que oferece como fatos (*facts*) os instrutores de cada classe (*instructor*) e em qual classe cada aluno está matriculado (*enrolled*). O programa usa esses fatos para responder quem é o instrutor de um estudante em particular. Esse programa deve usar os predicados *instructor*(p, c) e *enrolled*(s, c) para representar que o professor p é o instrutor do curso c e o estudante s está matriculado no curso c, respectivamente. Por exemplo, os *Prolog facts* nesse programa podem incluir:

```
instructor(chan,math273)
instructor(patel,ee222)
instructor(grossman,cs301)
enrolled(kevin,math273)
enrolled(juana,ee222)
enrolled(juana,cs301)
enrolled(kiko,math273)
enrolled(kiko,cs301)
```

(As letras minúsculas são usadas para as entradas, pois o Prolog considera nomes que começam por letras maiúsculas como variáveis.)

Um novo predicado *teaches*(p, s), que representa que o professor p ensina o estudante s, pode ser definido usando uma *Prolog rule*

```
teaches(P,S) :- instructor(P,C), enrolled(S,C)
```

o que significa que *teaches*(p, s) é verdadeiro se existir uma classe c tal que o professor p é instrutor dessa classe e o estudante s está matriculado na classe c. (Note que uma vírgula é usada para representar uma conjunção de predicados em Prolog. Similarmente, um ponto-e-vírgula é usado para representar uma disjunção de predicados.)

Prolog responde às perguntas usando os fatos e as regras dadas. Por exemplo, usando os fatos e as regras listadas, a pergunta

```
?enrolled(kevin,math273)
```

produz a resposta

```
yes
```

porque o fato *enrolled* (kevin, math273) foi dado como entrada. A pergunta

```
?enrolled(X,math273)
```

produz a resposta

```
kevin
kiko
```

Para produzir essa resposta, Prolog determina todos os valores possíveis da variável X, para os quais *enrolled*(X, math273) foi dado como *Prolog fact*. Similarmente, para encontrar os professores que são os instrutores das classes de Juana, usamos a pergunta

```
?teaches(X,juana)
```

Essa pergunta retorna

```
patel
grossman
```

◀

Exercícios

1. Considere $P(x)$ como a proposição "x $\leq$ 4". Quais são os valores-verdade das proposições abaixo?
   a) $P(0)$ **b)** $P(4)$ **c)** $P(6)$
2. Considere $P(x)$ como a proposição "a palavra x contém a letra a". Quais são os valores-verdade das proposições abaixo?
   a) P(orange) **b)** P(lemon)
   c) P(true) **d)** P(false)
3. Considere $Q(x, y)$ como a proposição "x é a capital de y." Quais os valores-verdade das proposições a seguir?
   a) Q(Denver, Colorado)
   b) Q(Detroit, Michigan)
   c) Q(Massachusetts, Boston)
   d) Q(Nova York, Nova York)
4. Constate o valor de x depois que a proposição **if** $P(x)$ **then** $x := 1$ for executada, em que $P(x)$ é a proposição "$x > 1$", se o valor de x, quando essa proposição for alcançada, for
   a) $x = 0$.
   b) $x = 1$.
   c) $x = 2$.
5. Considere $P(x)$ como a proposição "x passa mais do que cinco horas em aula todos os dias", em que o domínio de x são todos os estudantes. Expresse cada uma dessas quantificações em português.

a) $\exists x\, P(x)$ **b)** $\forall x\, P(x)$
c) $\exists x \neg P(x)$ **d)** $\forall x \neg P(x)$

6. Considere $N(x)$ como a proposição "x visitou Dakota do Norte", em que o domínio são os estudantes de sua escola. Expresse cada uma dessas quantificações em português.
a) $\exists x\, N(x)$ **b)** $\forall x\, N(x)$ **c)** $\neg\exists x\, N(x)$
d) $\exists x \neg N(x)$ **e)** $\neg\forall x\, N(x)$ **f)** $\forall x \neg N(x)$

7. Transcreva estas proposições para o português, em que $C(x)$ é "x é um comediante" e $F(x)$ é "x é divertido" e o domínio são todas as pessoas.
a) $\forall x(C(x) \rightarrow F(x))$ **b)** $\forall x(C(x) \wedge F(x))$
c) $\exists x(C(x) \rightarrow F(x))$ **d)** $\exists x(C(x) \wedge F(x))$

8. Transcreva estas proposições para o português, em que $R(x)$ é "x é um coelho" e $H(x)$ é "x salta" e o domínio são todos os animais.
a) $\forall x(R(x) \rightarrow H(x))$ **b)** $\forall x(R(x) \wedge H(x))$
c) $\exists x(R(x) \rightarrow H(x))$ **d)** $\exists x(R(x) \wedge H(x))$

9. Considere $P(x)$ como a proposição "x fala russo" e considere $Q(x)$ como a proposição "x sabe a linguagem computacional C++". Expresse cada uma dessas sentenças em termos de $P(x)$, $Q(x)$, quantificadores e conectivos lógicos. O domínio para quantificadores são todos os estudantes de sua escola.
a) Há um estudante em sua escola que fala russo e sabe C++.
b) Há um estudante em sua escola que fala russo, mas não sabe C++.
c) Todo estudante em sua escola ou fala russo ou sabe C++.
d) Nenhum estudante em sua escola fala russo ou sabe C++.

10. Considere $C(x)$ como a proposição "x tem um gato", $D(x)$ como "x tem um cachorro" e $F(x)$ como "x tem um furão". Expresse cada uma dessas proposições em termos de $C(x)$, $D(x)$, $F(x)$, quantificadores e conectivos lógicos. O domínio são todos os estudantes de sua sala.
a) Um estudante de sua sala tem um gato, um cachorro e um furão.
b) Todos os estudantes de sua sala têm um gato, um cachorro ou um furão.
c) Algum estudante de sua sala tem um gato e um furão, mas não tem um cachorro.
d) Nenhum estudante de sua sala tem um gato, um cachorro e um furão.
e) Para cada um desses três animais, gatos, cachorros e furões, há um estudante em sua sala que possui um dos três como animal de estimação.

11. Considere $P(x)$ como a proposição "$x = x^2$". Se o domínio forem os números inteiros, quais serão os valores-verdade?
a) $P(0)$ **b)** $P(1)$ **c)** $P(2)$
d) $P(-1)$ **e)** $\exists x P(x)$ **f)** $\forall x P(x)$

12. Considere $Q(x)$ como a proposição "$x + 1 > 2x$". Se o domínio forem todos os números inteiros, quais serão os valores- verdade?
a) $Q(0)$ **b)** $Q(-1)$ **c)** $Q(1)$
d) $\exists x Q(x)$ **e)** $\forall x Q(x)$ **f)** $\exists x \neg Q(x)$
g) $\forall x \neg Q(x)$

13. Determine o valor-verdade de cada uma destas proposições, se o domínio forem todos os números inteiros.
a) $\forall n(n + 1 > n)$ **b)** $\exists n(2n = 3n)$
c) $\exists n(n = -n)$ **d)** $\forall n(n^2 \geq n)$

14. Determine o valor-verdade de cada uma destas proposições, se o domínio forem todos os números reais.
a) $\exists x(x^3 = -1)$ **b)** $\exists x(x^4 < x^2)$
c) $\forall x((-x)^2 = x^2)$ **d)** $\forall x(2x > x)$

15. Determine o valor-verdade de cada uma destas proposições, se o domínio para todas as variáveis forem todos os números inteiros.
a) $\forall n(n^2 \geq 0)$ **b)** $\exists n(n^2 = 2)$
c) $\forall n(n^2 \geq n)$ **d)** $\exists n(n^2 < 0)$

16. Determine o valor-verdade de cada uma destas proposições, se o domínio de cada variável forem todos os números reais.
a) $\exists x(x^2 = 2)$ **b)** $\exists x(x^2 = -1)$
c) $\forall x(x^2 + 2 \geq 1)$ **d)** $\forall x(x^2 \neq x)$

17. Suponha que o domínio da função proposicional $P(x)$ sejam os números inteiros 0, 1, 2, 3 e 4. Desenvolva estas proposições usando disjunções, conjunções e negações.
a) $\exists x\, P(x)$ **b)** $\forall x\, P(x)$ **c)** $\exists x \neg P(x)$
d) $\forall x \neg P(x)$ **e)** $\neg\exists x\, P(x)$ **f)** $\neg\forall x\, P(x)$

18. Suponha que o domínio da função proposicional $P(x)$ sejam os números inteiros $-2, -1, 0, 1$ e 2. Desenvolva estas proposições usando disjunções, conjunções e negações.
a) $\exists x\, P(x)$ **b)** $\forall x\, P(x)$ **c)** $\exists x \neg P(x)$
d) $\forall x \neg P(x)$ **e)** $\neg\exists x\, P(x)$ **f)** $\neg\forall x\, P(x)$

19. Suponha que o domínio da função proposicional $P(x)$ sejam os números inteiros 1, 2, 3, 4 e 5. Expresse estas proposições sem usar quantificadores, mas, sim, apenas negações, disjunções e conjunções.
a) $\exists x\, P(x)$ **b)** $\forall x\, P(x)$
c) $\neg\exists x\, P(x)$ **d)** $\neg\forall x\, P(x)$
e) $\forall x\,((x \neq 3) \rightarrow P(x)) \vee \exists x \neg P(x)$

20. Suponha que o domínio da função proposicional $P(x)$ sejam $-5, -3, -1, 1, 3$ e 5. Expresse estas proposições sem usar quantificadores, mas, sim, apenas negações, disjunções e conjunções.
a) $\exists x\, P(x)$ **b)** $\forall x\, P(x)$
c) $\forall x((x \neq 1) \rightarrow P(x))$
d) $\exists x((x \geq 0) \wedge P(x))$
e) $\exists x\,(\neg P(x)) \wedge \forall x((x < 0) \rightarrow P(x))$

21. Para cada uma destas proposições, encontre um domínio para que a proposição seja verdadeira e um domínio para que a proposição seja falsa.
a) Todos estão estudando matemática discreta.
b) Todos têm mais de 21 anos.
c) Duas pessoas têm a mesma mãe.
d) Nem todo par de pessoas diferentes tem a mesma avó.

22. Para cada uma destas proposições, encontre um domínio para que a proposição seja verdadeira e um domínio para que a proposição seja falsa.
a) Todos falam hindu.
b) Há alguém com mais de 21 anos.

c) Cada duas pessoas têm o mesmo nome.
d) Alguém sabe mais que duas outras pessoas.

23. Transcreva de duas formas cada uma das proposições em expressões lógicas usando predicados, quantificadores e conectivos lógicos. Primeiro, o domínio são os estudantes em sua sala, e, segundo, considere-o como todas as pessoas.
a) Alguém em sua sala fala hindu.
b) Todos em sua sala são amigáveis.
c) Há uma pessoa em sua sala que não nasceu na Califórnia.
d) Um estudante de sua sala participou de um filme.
e) Nenhum estudante de sua sala teve um curso de programação lógica.

24. Transcreva de duas formas cada uma das proposições em expressões lógicas usando predicados, quantificadores e conectivos lógicos. Primeiro, o domínio são os estudantes em sua sala, e, segundo, considere-o como todas as pessoas.
a) Todos em sua sala têm um celular.
b) Alguém em sua sala viu um filme estrangeiro.
c) Há uma pessoa em sua sala que não sabe nadar.
d) Todos os estudantes em sua sala sabem resolver equações quadráticas.
e) Algum estudante de sua sala não quer ficar rico.

25. Transcreva cada uma das proposições em expressões lógicas usando predicados, quantificadores e conectivos lógicos.
a) Ninguém é perfeito.
b) Nem todos são perfeitos.
c) Todos os seus amigos são perfeitos.
d) Pelo menos um de seus amigos é perfeito.
e) Todos são seus amigos e são perfeitos.
f) Nem todos são seus amigos ou alguém não é perfeito.

26. Transcreva cada uma das proposições abaixo em expressões lógicas de três maneiras diferentes, variando o domínio e usando predicados com uma ou duas variáveis.
a) Alguém em sua escola visitou o Uzbequistão.
b) Todos em sua sala estudaram cálculo e C++.
c) Ninguém em sua escola tem uma bicicleta e uma moto.
d) Há uma pessoa em sua escola que não é feliz.
e) Todos em sua escola nasceram no século XX.

27. Transcreva cada uma das proposições abaixo em expressões lógicas de três maneiras diferentes, variando o domínio e usando predicados com uma ou duas variáveis.
a) Um estudante em sua escola morou no Vietnã.
b) Há um estudante em sua escola que não fala hindu.
c) Um estudante em sua escola conhece Java, Prolog e C++.
d) Todos em sua sala gostam de comida tailandesa.
e) Alguém em sua sala não joga hóquei.

28. Transcreva as proposições a seguir em expressões lógicas, usando predicados, quantificadores e conectivos lógicos.
a) Alguma coisa não está no lugar certo.
b) Todos os instrumentos estão no lugar certo e em excelentes condições.
c) Tudo está no lugar certo e em excelente condição.
d) Nada está no lugar certo e em excelente condição.
e) Um de seus instrumentos não está no lugar correto, mas está em excelente condição.

29. Expresse cada uma das proposições abaixo usando operadores lógicos, predicados e quantificadores.
a) Algumas proposições são tautologias.
b) A negação de uma contradição é uma tautologia.
c) A disjunção de duas contingências pode ser uma tautologia.
d) A conjunção de duas tautologias é uma tautologia.

30. Suponha que o domínio de uma função proposicional $P(x, y)$ sejam pares de x e y, em que x é 1, 2 ou 3 e y é 1, 2 ou 3. Desenvolva as proposições abaixo usando disjunções e conjunções.
a) $\exists x\, P(x, 3)$ **b)** $\forall y\, P(1, y)$
c) $\exists y \neg P(2, y)$ **d)** $\forall x \neg P(x, 2)$

31. Suponha que o domínio de $Q(x, y, z)$ sejam as três variáveis x, y, z, em que $x = 0$, 1 ou 2, $y = 0$ ou 1 e $z = 0$ ou 1. Desenvolva as proposições abaixo usando disjunções e conjunções.
a) $\forall y\, Q(0, y, 0)$ **b)** $\exists x\, Q(x, 1, 1)$
c) $\exists z \neg Q(0, 0, z)$ **d)** $\exists x \neg Q(x, 0, 1)$

32. Expresse cada uma das proposições abaixo usando quantificadores. Depois, forme a negação da proposição; nenhuma negação pode ficar do lado esquerdo de um quantificador. Em seguida, expresse a negação em português. (Não use simplesmente as palavras "Não é o caso de".)
a) Todos os cães têm pulgas.
b) Há um cavalo que trota.
c) Todo coala sabe escalar.
d) Nenhum macaco fala francês.
e) Lá existe um porco que nada e caça peixes.

33. Expresse cada uma das proposições abaixo usando quantificadores. Depois, forme a negação da proposição; nenhuma negação pode ficar do lado esquerdo de um quantificador. Em seguida, expresse a negação em português. (Não use simplesmente as palavras "Não é o caso de".)
a) Alguns cães velhos aprendem truques novos.
b) Nenhum coelho sabe cálculo.
c) Todo pássaro pode voar.
d) Não há cães que falem.
e) Não há nesta sala alguém que fale francês e russo.

34. Expresse a negação das proposições abaixo usando quantificadores e depois expresse a negação em português.
a) Alguns motoristas não obedecem aos limites de velocidade.
b) Todos os filmes suíços são sérios.
c) Ninguém pode guardar um segredo.
d) Há alguém nesta sala que não tem uma boa atitude.

35. Encontre um contra-exemplo, se possível, para estas proposições quantificadas universalmente, em que o domínio para todas as variáveis são todos os números inteiros.
a) $\forall x(x^2 \geq x)$
b) $\forall x(x > 0 \vee x < 0)$
c) $\forall x(x = 1)$

36. Encontre um contra-exemplo, se possível, para estas proposições quantificadas universalmente, em que o domínio para todas as variáveis são todos os números reais.
a) $\forall x(x^2 \neq x)$ **b)** $\forall x(x^2 \neq 2)$
c) $\forall x(|x| > 0)$

37. Expresse cada uma das proposições abaixo usando predicados e quantificadores.
a) Um passageiro em uma companhia aérea é qualificado como um viajante de elite se voar mais de 25.000 milhas em um ano ou pegar mais de 25 vôos durante o ano.
b) Um homem se classifica para a maratona se seu melhor tempo for menor que 3 horas, e uma mulher, se seu melhor tempo for menor que 3,5 horas.
c) Um estudante deve freqüentar no mínimo 60 horas/aula, ou pelo menos 45 horas/aula e escrever uma tese, e ter, em todas as disciplinas, conceito não menor que B para receber o título de mestre.
d) Há um estudante que cursou mais de 21 créditos em um semestre e recebeu apenas conceitos A.

Os exercícios 38 a 42 lidam com a transcrição entre sistema de especificação e expressões lógicas que envolvem quantificadores.

38. Transcreva os sistemas de especificações abaixo para o português, em que o predicado $S(x, y)$ é "x está em estado y" e o domínio para x e y são todos os sistemas e todos os estados possíveis, respectivamente.
a) $\exists x\, S(x,$ aberto$)$
b) $\forall x\,(S(x,$ em mau funcionamento$) \vee S(x,$ diagnóstico$))$
c) $\exists x\, S(x,$ aberto$) \vee \exists x\, S(x,$ diagnóstico$)$
d) $\exists x \neg S(x,$ disponível$)$
e) $\forall x \neg S(x,$ em funcionamento$)$

39. Transcreva as especificações abaixo para o português, em que $F(p)$ é "A impressora p está quebrada", $B(p)$ é "A impressora p está ocupada", $L(j)$ é "A impressão do trabalho j foi perdida" e $Q(j)$ é "A impressão do trabalho j foi adicionada à fila".
a) $\exists p(F(p) \wedge B(p)) \rightarrow \exists j L(j)$
b) $\forall p B(p) \rightarrow \exists j Q(j)$
c) $\exists j(Q(j) \wedge L(j)) \rightarrow \exists p F(p)$
d) $(\forall p B(p) \wedge \forall j Q(j)) \rightarrow \exists j L(j)$

40. Expresse cada um dos sistemas de especificações a seguir, usando predicados, quantificadores e conectivos lógicos.
a) Quando há menos de 30 megabytes livres no disco rígido, um aviso é enviado a todos os usuários.
b) Nenhum diretório no sistema de arquivos pode ser aberto e nenhum arquivo pode ser fechado se forem detectados erros no sistema.
c) O sistema de arquivos não pode ser recuperado se houver um usuário conectado.
d) Vídeos para downloads podem ser baixados se houver pelo menos 8 megabytes de memória disponíveis e a velocidade de conexão for, no mínimo, de 56 kilobits por segundo.

41. Expresse cada um dos sistemas de especificações abaixo, usando predicados, quantificadores e conectivos lógicos.
a) Pelo menos um e-mail, entre o conjunto total de mensagens, pode ser salvo se o disco estiver com mais de 10 kilobytes de espaço livre.
b) Sempre que o alerta estiver ativado, todas as mensagens na fila serão transmitidas.
c) A tela de diagnóstico rastreia o *status* de todos os sistemas, exceto do console principal.
d) Cada participante da videoconferência que o anfitrião da chamada não pôs em uma lista especial receberá uma conta.

42. Expresse cada um dos sistemas de especificações abaixo, usando predicados, quantificadores e conectivos lógicos.
a) Todo usuário tem acesso a uma caixa de entrada.
b) O sistema de caixa de entrada pode ser acessado por qualquer pessoa no grupo se o sistema for travado.
c) O firewall está em estado de diagnóstico apenas se o servidor de proxy estiver em estado de diagnóstico.
d) Pelo menos um roteador funcionará normalmente se a entrada for entre 100 kbps e 500 kbps e o servidor de proxy não estiver em modo de diagnóstico.

43. Determine se $\forall x(P(x) \rightarrow Q(x))$ e $\forall x\, P(x) \rightarrow \forall x\, Q(x)$ são logicamente equivalentes. Justifique sua resposta.

44. Determine se $\forall x(P(x) \leftrightarrow Q(x))$ e $\forall x\, P(x) \leftrightarrow \forall x\, Q(x)$ são logicamente equivalentes. Justifique sua resposta.

45. Mostre que $\exists x\,(P(x) \vee Q(x))$ e $\exists x\,(P(x) \vee \exists x Q(x)$ são logicamente equivalentes.

Os exercícios 46 a 49 estabelecem regras para a **quantificação nula** que pode ser usada quando uma variável quantificadora não aparece em parte de uma proposição.

46. Estabeleça as equivalências lógicas abaixo, em que x não aparece como variável livre em A. Assuma que o domínio não é vazio.
a) $(\forall x\, P(x)) \vee A \equiv \forall x(P(x) \vee A)$
b) $(\exists x\, P(x)) \vee A \equiv \exists x(P(x) \vee A)$

47. Estabeleça as equivalências lógicas abaixo, em que x não aparece como variável livre em A. Assuma que o domínio não é vazio.
a) $(\forall x\, P(x)) \wedge A \equiv \forall x(P(x) \wedge A)$
b) $(\exists x\, P(x)) \wedge A \equiv \exists x(P(x) \wedge A)$

48. Estabeleça as equivalências lógicas abaixo, em que x não aparece como uma variável livre em A. Assuma que o domínio não é vazio.
a) $\forall x\,(A \rightarrow P(x)) \equiv A \rightarrow \forall x\, P(x)$
b) $\exists x\,(A \rightarrow P(x)) \equiv A \rightarrow \exists x\, P(x)$

49. Estabeleça as equivalências lógicas abaixo, em que x não aparece como variável livre em A. Assuma que o domínio não é vazio.
a) $\forall x\,(P(x) \rightarrow A) \equiv \exists x\, P(x) \rightarrow A$
b) $\exists x\,(P(x) \rightarrow A) \equiv \forall x\, P(x) \rightarrow A$

50. Mostre que $\forall x\, P(x) \vee \forall x\, Q(x)$ e $\forall x(P(x) \vee Q(x))$ não são logicamente equivalentes.

51. Mostre que $\exists x\, P(x) \wedge \exists x\, Q(x)$ e $\exists x\,(P(x) \wedge Q(x))$ não são logicamente equivalentes.

52. Como mencionado no texto, a notação $\exists! x\, P(x)$ refere-se a

"Existe um único x para que $P(x)$ seja verdadeira."

Se o domínio forem todos os números inteiros, quais serão os valores-verdade das proposições abaixo?

a) $\exists! x\,(x > 1)$ **b)** $\exists! x\,(x^2 = 1)$

c) $\exists! x(x + 3 = 2x)$ **d)** $\exists! x(x = x + 1)$

53. Quais são os valores-verdade das proposições abaixo?

a) $\exists! x\, P(x) \rightarrow \exists x\, P(x)$

b) $\forall x\, P(x) \rightarrow \exists! x\, P(x)$

c) $\exists! x \neg P(x) \rightarrow \neg \forall x\, P(x)$

54. Desenvolva $\exists! x P(x)$, em que o domínio são os números inteiros 1, 2 e 3, em termos de negação, conjunção e disjunção.

55. Dados os fatos de Prolog no Exemplo 28, qual seria o retorno de Prolog dado nestas perguntas?

a) `?instructor(chan,math273)`

b) `?instructor(patel,cs301)`

c) `?enrolled(X,cs301)`

d) `?enrolled(kiko,Y)`

e) `?teaches(grossman,Y)`

56. Dados os fatos de Prolog no Exemplo 28, qual seria o retorno de Prolog dado nestas perguntas?

a) `?enrolled(kevin,ee222)`

b) `?enrolled(kiko,math273)`

c) `?instructor(grossman,X)`

d) `?instructor(X,cs301)`

e) `?teaches(X,kevin)`

57. Suponha que os fatos de Prolog sejam usados para definir os predicados *mãe*(M, Y) e *pai*(F, X), em que M representa a mãe de Y e F é o pai de X, respectivamente. Dê uma regra de Prolog para definir o predicado *irmão*(X, Y), em que X e Y são irmãos (ou seja, têm o mesmo pai e a mesma mãe).

58. Suponha que os fatos de Prolog sejam usados para definir os predicados *mãe*(M, Y) e *pai*(F, X), em que M representa a mãe de Y e F é o pai de X, respectivamente. Dê uma regra de Prolog para definir o predicado *avô*(X, Y), que representa que X é o avô de Y. [*Dica:* Você pode desenvolver a disjunção no Prolog ou usando um ponto-e-vírgula para separar predicados ou pondo esses predicados em linhas separadas.]

Os exercícios 59 a 62 têm como base questões encontradas no livro *Lógica Simbólica,* de Lewis Carroll.

59. Considere $P(x)$, $Q(x)$ e $R(x)$ como as proposições "x é um professor", "x é ignorante" e "x é vão", respectivamente. Expresse cada uma das proposições usando quantificadores, conectivos lógicos e $P(x)$, $Q(x)$ e $R(x)$, em que o domínio são todas as pessoas.

a) Nenhum professor é ignorante.

b) Todas as pessoas ignorantes são vãs.

c) Nenhum professor é vão.

d) O item (c) resulta de (a) e (b)?

60. Considere $P(x)$, $Q(x)$ e $R(x)$ como as proposições "x é uma explicação clara", "x é satisfatório" e "x é uma desculpa", respectivamente. Suponha que o domínio de x seja todo o texto em português. Expresse cada uma das proposições abaixo usando quantificadores, conectivos lógicos e $P(x)$, $Q(x)$ e $R(x)$.

a) Todas as explicações claras são satisfatórias.

b) Todas as desculpas não são satisfatórias.

c) Algumas desculpas não são explicações claras.

***d)** O item (c) resulta de (a) e (b)?

61. Considere $P(x)$, $Q(x)$, $R(x)$ e $S(x)$ como as proposições "x é um bebê", "x é lógico", "x é capaz de controlar um crocodilo" e "x é desprezível", respectivamente. Suponha que o domínio sejam todas as pessoas. Expresse cada uma das proposições abaixo usando quantificadores, conectivos lógicos e $P(x)$, $Q(x)$, $R(x)$ e $S(x)$.

a) Bebês não são lógicos.

b) Ninguém é desprezível se pode controlar um crocodilo.

c) Pessoas que não são lógicas são desprezíveis.

d) Bebês não podem controlar crocodilos.

***e)** O item (d) resulta de (a), (b) e (c)? Se não, existe alguma conclusão correta?

62. Considere $P(x)$, $Q(x)$, $R(x)$ e $S(x)$ como as proposições "x é um pato", "x é uma das minhas aves", "x é um oficial" e "x está valsando", respectivamente. Expresse cada uma das proposições abaixo usando quantificadores, conectivos lógicos e $P(x)$, $Q(x)$, $R(x)$ e $S(x)$.

a) Nenhum pato está valsando.

b) Nenhum oficial nunca recusa uma valsa.

c) Todas as minhas aves são patos.

d) Minhas aves não são oficiais.

***e)** O item (d) resulta de (a), (b) e (c)? Se não, há uma conclusão correta?

1.4 Quantificadores Agrupados

Introdução

Na Seção 1.3, definimos os quantificadores existencial e universal e mostramos como podem ser usados em sentenças matemáticas. Também explicamos como podem ser usados para traduzir

sentenças do português para expressões lógicas. Nesta seção, vamos estudar **quantificadores agrupados**. Dois quantificadores são agrupados se um está no escopo do outro, tal como

$$\forall x \exists y (x + y = 0).$$

Note que tudo que está no escopo de um quantificador pode ser considerado uma função proposicional. Por exemplo,

$$\forall x \exists y (x + y = 0)$$

é o mesmo que $\forall x Q(x)$, em que $Q(x)$ é $\exists y P(x, y)$ e $P(x, y)$ é $x + y = 0$. Quantificadores agrupados sempre ocorrem em matemática e ciência da computação. No entanto, quantificadores agrupados podem ser, às vezes, difíceis de entender; as regras que estudamos na Seção 1.3 podem nos ajudar.

Para entender essas sentenças que envolvem muitos quantificadores, precisamos esclarecer o que significa cada predicado e cada quantificador que aparece. Isso é ilustrado nos exemplos 1 e 2.

EXEMPLO 1 Assuma que o domínio para as variáveis x e y consiste em todos os números reais. A sentença

$$\forall x \forall y (x + y = y + x)$$

Exemplos Extras diz que $x + y = y + x$ para todos os números reais x e y. Essa é a propriedade comutativa para números reais. De maneira análoga, a sentença

$$\forall x \exists y (x + y = 0)$$

diz que para cada (para todo) número real x existe um número real y, tal que $x + y = 0$. Essa sentença diz que todo número real tem um inverso aditivo (o oposto). Similarmente, a sentença

$$\forall x \forall y \forall z (x + (y + z) = (x + y) + z)$$

é a propriedade associativa da adição para números reais. ◀

EXEMPLO 2 Traduza para o português a sentença

$$\forall x \forall y ((x > 0) \wedge (y < 0) \rightarrow (xy < 0)),$$

em que o domínio para ambas as variáveis são números reais.

Solução: Essa sentença diz que para todo número real x e para todo número real y, se $x > 0$ e $y < 0$, então $xy < 0$. Ou, ainda, para números reais x e y, se x é positivo e y é negativo, então xy é negativo. Isso pode ser descrito mais sucintamente por "O produto de um número positivo por um número negativo é sempre negativo". ◀

PENSANDO EM QUANTIFICAÇÕES COMO LAÇOS Trabalhar com quantificações de mais de uma variável às vezes facilita o fato de pensarmos em termos de laços agrupados. (É claro que, se existirem infinitos elementos no domínio de alguma variável, não podemos dar uma volta por todos os valores. No entanto, essa maneira de pensar pode nos ajudar a entender quantificadores agrupados.) Por exemplo, para ver se $\forall x \forall y P(x, y)$ é verdadeira, podemos dar uma passada por todos os valores de x, e para cada valor de x, passamos por todos os valores de y. Se encontrarmos $P(x, y)$ verdadeira para todos os valores de x e y, teremos determinado que $\forall x \forall y P(x, y)$ é verdadeira. Se chegarmos a algum valor de x para o qual encontramos algum valor de y, tal que $P(x, y)$ seja falsa, teremos mostrado que $\forall x \forall y P(x, y)$ é falsa.

Similarmente, para determinar quando $\forall x \exists y P(x, y)$ é verdadeira, damos uma volta por todos os valores de x. Para cada valor de x, damos uma volta sobre os valores de y até achar um valor

para o qual $P(x, y)$ é verdadeira. Se para todo valor de x encontrarmos um y, então $\forall x \exists y P(x, y)$ é verdadeira; se para algum x nunca encontrarmos um valor de y, então $\forall x \exists y P(x, y)$ é falsa.

Para ver se $\exists x \forall y P(x, y)$ é verdadeira, podemos dar uma volta pelos valores de x até encontrarmos um valor de x para o qual $P(x, y)$ é sempre verdadeira quando damos uma volta pelos valores de y. Se encontrarmos tal x, saberemos que $\exists x \forall y P(x, y)$ é verdadeira. Se não encontrarmos tal x, concluiremos que $\exists x \forall y P(x, y)$ é falsa.

Finalmente, para ver se $\exists x \exists y P(x, y)$ é verdadeira, damos uma volta pelos valores de x, em que para cada x damos uma volta pelos valores de y até encontrarmos um x para o qual existe um y, tal que $P(x, y)$ seja verdadeira. A sentença $\exists x \exists y P(x, y)$ será falsa somente se não encontrarmos um x para o qual encontramos algum y tal que $P(x, y)$ seja verdadeira.

A Ordem dos Quantificadores

Muitas sentenças matemáticas envolvem múltiplas quantificações de funções proposicionais com mais de uma variável. É importante notar que a ordem dos quantificadores precisa ser considerada, a menos que os quantificadores não sejam todos universais ou todos existenciais.

Essa observação é ilustrada pelos exemplos 3 a 5.

EXEMPLO 3 Seja $P(x, y)$ a sentença "$x + y = y + x$". Quais os valores-verdade das quantificações $\forall x \forall y P(x, y)$ e $\forall y \forall x P(x, y)$, em que o domínio para as variáveis consiste em todos os números reais?

Solução: A quantificação

$$\forall x \forall y P(x, y)$$

Exemplos Extras

indica a proposição

"Para todo número real x, para todo número real y, $x + y = y + x$."

Como $P(x, y)$ é verdadeira para todos os números reais x e y (esta é a propriedade comutativa para a adição, que é um axioma para todos os números reais — veja o Apêndice 1), a proposição $\forall x \forall y P(x, y)$ é verdadeira. Note que a sentença $\forall y \forall x P(x, y)$ diz "Para todo número real y, para todo número real x, $x + y = y + x$". Isto tem o mesmo significado que a sentença "Para todo número real x, para todo número real y, $x + y = y + x$". Ou seja, $\forall x \forall y P(x, y)$ e $\forall y \forall x P(x, y)$ têm o mesmo significado, e são ambas verdadeiras. Isso ilustra o princípio que diz que a ordem dos quantificadores universais em uma sentença sem outros quantificadores pode ser mudada sem alterar o significado da sentença quantificada. ◀

EXEMPLO 4 Seja $Q(x, y)$ a sentença "$x + y = 0$". Quais os valores-verdade das quantificações $\exists y \forall x Q(x, y)$ e $\forall x \exists y Q(x, y)$, em que o domínio para as variáveis consiste em todos os números reais?

Solução: A quantificação

$$\exists y \forall x Q(x,y)$$

indica a proposição

"Existe um número real y para todo número real x, $Q(x, y)$."

Qualquer que seja o valor de y escolhido, existe um valor x para o qual $x + y = 0$. Como não existe número real y tal que $x + y = 0$ para todo número real x, a sentença $\exists y \forall x Q(x, y)$ é falsa.

TABELA 1 Quantificações de Duas Variáveis.

Sentença	*Quando é verdadeira?*	*Quando é falsa?*
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ é verdadeira para todo par x, y.	Existe um par x, y para o qual $P(x, y)$ é falsa.
$\forall x \exists y P(x, y)$	Para todo x existe um y para o qual $P(x, y)$ é verdadeira.	Existe um x tal que $P(x, y)$ é falsa para todo y.
$\exists x \forall y P(x, y)$	Existe um x tal que $P(x, y)$ é verdadeira para todo y.	Para todo x existe um y para o qual $P(x, y)$ é falsa.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	Existe um par x, y para o qual $P(x, y)$ é verdadeira.	$P(x, y)$ é falsa para todo par x, y.

A quantificação

$$\forall x \exists y Q(x, y)$$

indica a proposição

"Para todo número real x existe um número real y tal que $Q(x, y)$."

Dado um número real x, existe um número real y tal que $x + y = 0$; isto é, $y = -x$. Portanto, $\forall x \exists y Q(x, y)$ é verdadeira. ◀

O Exemplo 4 ilustra que a ordem em que aparecem os quantificadores faz muita diferença. As sentenças $\exists y \forall x Q(x, y)$ e $\forall x \exists y Q(x, y)$ não são logicamente equivalentes. A sentença $\exists y \forall x Q(x, y)$ é verdadeira se e somente se existe um y que faz com que $Q(x, y)$ seja verdadeira para todo x. Portanto, para essa sentença ser verdadeira, deve existir um valor particular de y para o qual $Q(x, y)$ é verdadeira independentemente da escolha de x. Por outro lado, $\forall x \exists y Q(x, y)$ é verdadeira se para todo x existe um número y para o qual $Q(x, y)$ é verdadeira. Portanto, para essa sentença ser verdadeira, qualquer que seja o valor de x que você escolha, deve existir um valor de y (possivelmente dependente do valor de x escolhido) para o qual $Q(x, y)$ é verdadeira. Em outras palavras, no segundo caso, y pode depender de x, enquanto, no primeiro caso, y é uma constante independente de x.

Dessas observações, segue-se que $\exists y \forall x Q(x, y)$ é verdadeira, então $\forall x \exists y Q(x, y)$ deve ser verdadeira. No entanto, se $\forall x \exists y Q(x, y)$ é verdadeira, não necessariamente $\exists y \forall x Q(x, y)$ deve ser verdadeira. (Veja os exercícios suplementares 24 e 25 no final deste capítulo.)

A Tabela 1 resume o significado das possíveis quantificações diferentes que envolvem duas variáveis.

Quantificações com mais de duas variáveis também são comuns, como ilustra o Exemplo 5.

EXEMPLO 5 Seja $Q(x, y, z)$ a sentença "$x + y = z$". Quais os valores-verdade das sentenças $\forall x \forall y \exists z Q(x, y, z)$ e $\exists z \forall x \forall y Q(x, y, z)$, em que o domínio consiste em todos os números reais?

Solução: Suponha que x e y sejam valores determinados. Então, existe um número real z tal que $x + y = z$. Conseqüentemente, a quantificação

$$\forall x \forall y \exists z Q(x, y, z),$$

EXEMPLO 8 (*Requer cálculo*) Expresse a definição de limite usando quantificadores.

Solução: Lembremos a definição de limite

$$\lim_{x \to a} f(x) = L$$

é: para todo real $\epsilon > 0$ existe um número real $\delta > 0$ tal que $|f(x) - L| < \epsilon$ toda vez que $0 < |x - a| < \delta$. Essa definição de limite pode ser expressa em termos de quantificadores por

$$\forall \epsilon \, \exists \, \delta \, \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon),$$

em que o domínio das variáveis δ e ϵ consiste em todos os reais, e, para x, o domínio são todos os reais.

Essa definição também pode ser expressa por

$$\forall \epsilon > 0 \, \exists \delta > 0 \, \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon)$$

quando o domínio das variáveis ϵ e δ consiste em todos os reais, em vez de apenas os números reais positivos. [Aqui, quantificadores restritos devem ser utilizados. Lembre que $\forall x > 0 \, P(x)$ significa que para todo x com $x > 0$, $P(x)$ é verdadeira.] ◀

Traduzindo Quantificadores Agrupados para o Português

Pode ser complicado traduzir expressões com quantificadores agrupados para sentenças em português (ou qualquer outra língua). O primeiro passo na tradução é escrever por extenso o que os quantificadores e predicados da expressão significam. O próximo passo é expressar esse significado em uma sentença o mais simples possível. Esse processo é ilustrado nos exemplos 9 e 10.

EXEMPLO 9 Traduza a sentença

$$\forall x (C(x) \vee \exists y (C(y) \wedge F(x, y)))$$

para o português, em que $C(x)$ é "x tem um computador", $F(x, y)$ é "x e y são amigos" e o domínio para ambas as variáveis são os estudantes da sua escola.

Solução: A sentença diz que para todo estudante x de sua escola, x tem um computador ou existe um estudante y tal que y tem um computador e x e y são amigos. Em outras palavras, todo estudante da sua escola tem um computador ou tem um amigo que tem um computador. ◀

EXEMPLO 10 Traduza a sentença

$$\exists x \forall y \forall z ((F(x, y) \wedge F(x, z) \wedge (y \neq z)) \rightarrow \neg (F(y, z))$$

para o português, em que $F(x, y)$ é "x e y são amigos" e o domínio para as variáveis x, y e z são todos os estudantes da sua escola.

Solução: Primeiro examinemos a expressão $(F(x, y) \wedge F(x, z) \wedge (y \neq z)) \rightarrow \neg F(y, z)$. Ela diz que se os estudantes x e y são amigos e os estudantes x e z são amigos e, ainda mais, se y e z não são o mesmo estudante, então y e z não são amigos. Daqui segue que a sentença original, que é triplamente quantificada, diz que existe um estudante x tal que para todo estudante y e para todo estudante z diferente de y, se x e y são amigos e x e z são amigos, então y e z não são amigos. Em outras palavras, existe um estudante, tal que nenhum par de amigos seus é também um par de amigos entre si. ◀

EXEMPLO 8 (*Requer cálculo*) Expresse a definição de limite usando quantificadores.

Solução: Lembremos a definição de limite

$$\lim_{x \to a} f(x) = L$$

é: para todo real $\epsilon > 0$ existe um número real $\delta > 0$ tal que $|f(x) - L| < \epsilon$ toda vez que $0 < |x - a| < \delta$. Essa definição de limite pode ser expressa em termos de quantificadores por

$$\forall \epsilon \, \exists \, \delta \, \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon),$$

em que o domínio das variáveis δ e ϵ consiste em todos os reais, e, para x, o domínio são todos os reais.

Essa definição também pode ser expressa por

$$\forall \epsilon > 0 \, \exists \delta > 0 \, \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon)$$

quando o domínio das variáveis ϵ e δ consiste em todos os reais, em vez de apenas os números reais positivos. [Aqui, quantificadores restritos devem ser utilizados. Lembre que $\forall x > 0 \, P(x)$ significa que para todo x com $x > 0$, $P(x)$ é verdadeira.] ◀

Traduzindo Quantificadores Agrupados para o Português

Pode ser complicado traduzir expressões com quantificadores agrupados para sentenças em português (ou qualquer outra língua). O primeiro passo na tradução é escrever por extenso o que os quantificadores e predicados da expressão significam. O próximo passo é expressar esse significado em uma sentença o mais simples possível. Esse processo é ilustrado nos exemplos 9 e 10.

EXEMPLO 9 Traduza a sentença

$$\forall x (C(x) \vee \exists y (C(y) \wedge F(x, y)))$$

para o português, em que $C(x)$ é "x tem um computador", $F(x, y)$ é "x e y são amigos" e o domínio para ambas as variáveis são os estudantes da sua escola.

Solução: A sentença diz que para todo estudante x de sua escola, x tem um computador ou existe um estudante y tal que y tem um computador e x e y são amigos. Em outras palavras, todo estudante da sua escola tem um computador ou tem um amigo que tem um computador. ◀

EXEMPLO 10 Traduza a sentença

$$\exists x \forall y \forall z ((F(x, y) \wedge F(x, z) \wedge (y \neq z)) \rightarrow \neg (F(y, z))$$

para o português, em que $F(x, y)$ é "x e y são amigos" e o domínio para as variáveis x, y e z são todos os estudantes da sua escola.

Solução: Primeiro examinemos a expressão $(F(x, y) \wedge F(x, z) \wedge (y \neq z)) \rightarrow \neg F(y, z)$. Ela diz que se os estudantes x e y são amigos e os estudantes x e z são amigos e, ainda mais, se y e z não são o mesmo estudante, então y e z não são amigos. Daqui segue que a sentença original, que é triplamente quantificada, diz que existe um estudante x tal que para todo estudante y e para todo estudante z diferente de y, se x e y são amigos e x e z são amigos, então y e z não são amigos. Em outras palavras, existe um estudante, tal que nenhum par de amigos seus é também um par de amigos entre si. ◀

Traduzindo Sentenças do Português para Expressões Lógicas

Na Seção 1.3 mostramos como quantificadores podem ser utilizados para traduzir sentenças para expressões lógicas. Contudo, queremos analisar sentenças tal que suas traduções para expressões lógicas requeiram o uso de quantificadores agrupados. Vamos agora trabalhar com traduções dessas sentenças.

EXEMPLO 11 Expresse a sentença "Se uma pessoa é do sexo feminino e tem filhos, então ela é mãe de alguém" como uma expressão lógica que envolve predicados, quantificadores com domínio que consiste em todas as pessoas e conectivos lógicos.

Solução: A sentença "Se uma pessoa é do sexo feminino e tem filhos, então ela é mãe de alguém" pode ser reescrita por "Para toda pessoa x, se x é do sexo feminino e x tem filhos, então existe uma pessoa y tal que x é mãe de y". Introduziremos a função proposicional $F(x)$ para representar "x é do sexo feminino", $P(x)$ para representar "x tem filhos" e $M(x, y)$ para representar "x é mãe de y". Então a sentença original pode ser escrita por

$$\forall x((F(x) \wedge P(x)) \rightarrow \exists y M(x, y)).$$

Usando a regra da quantificação nula na parte (b) do Exercício 47 da Seção 1.3, podemos mover $\exists y$ para a esquerda e esse quantificador aparecerá depois de $\forall x$, porque y não aparece em $F(x) \wedge P(x)$. Então obtemos a expressão logicamente equivalente

$$\forall x \exists y((F(x) \wedge P(x)) \rightarrow M(x, y)).$$ ◀

EXEMPLO 12 Expresse a sentença "Todos têm exatamente um melhor amigo" como uma expressão lógica que envolve predicados, quantificadores com domínio que consiste em todas as pessoas e conectivos lógicos.

Solução: A sentença "Todos têm exatamente um melhor amigo" pode ser expressa por "Para toda pessoa x, x tem exatamente um melhor amigo". Introduzindo o quantificador universal, vemos que essa sentença é o mesmo que "$\forall x$ (x tem exatamente um melhor amigo)", em que o domínio consiste em todas as pessoas.

Dizer que x tem exatamente um melhor amigo é o mesmo que dizer que existe uma pessoa y que é o melhor amigo de x, e, mais ainda, que para toda pessoa z, se z não é y, então z não é o melhor amigo de x. Quando introduzimos o predicado $B(x, y)$ para representar "y é o melhor amigo de x", a sentença "x tem exatamente um melhor amigo" pode ser representada por

$$\exists y(B(x, y) \wedge \forall z((z \neq y) \rightarrow \neg B(x, z))).$$

Conseqüentemente, nossa sentença original pode ser expressa por

$$\forall x \exists y(B(x, y) \wedge \forall z((z \neq y) \rightarrow \neg B(x, z))).$$

[Note que poderíamos escrever esta sentença como $\forall x \exists! y B(x, y)$ em que $\exists!$ é o "quantificador de unicidade" definido na página 37.] ◀

EXEMPLO 13 Use quantificadores para expressar a sentença "Existe uma mulher que já tomou um avião em todas as linhas aéreas do mundo".

Solução: Sejam $P(w,f)$ o predicado "w tomou o avião f" e $Q(f,a)$ o predicado "f é um avião da linha a". Podemos expressar a sentença como

$$\exists w \forall a \exists f (P(w,f) \wedge Q(f,a)),$$

em que o domínio de discurso para w, f e a consiste em todas as mulheres do mundo, todos aviões e todas as linhas aéreas, respectivamente.

A sentença também pode ser expressa por

$$\exists w \forall a \exists f \; R(w,f,a),$$

em que $R(w,f,a)$ é "w tomou o avião f na linha a". Embora essa seja mais compacta, é um pouco obscura a relação entre as variáveis. Conseqüentemente, a primeira solução é preferível. ◀

Negando Quantificadores Agrupados

Auto-avaliação Sentenças que envolvem quantificadores agrupados podem ser negadas por aplicações sucessivas das regras de negação de sentenças com um único quantificador. Isso é ilustrado nos exemplos 14 a 16.

EXEMPLO 14 Expresse a negação da sentença $\forall x \exists y (xy = 1)$ de tal forma que a negação não preceda algum quantificador.

Exemplos Extras *Solução*: Por sucessivas aplicações das leis de De Morgan para quantificadores na Tabela 2 da Seção 1.3, podemos mover a negação em $\neg \forall x \exists y (xy = 1)$ para dentro de todos os quantificadores. Vemos que $\neg \forall x \exists y (xy = 1)$ é equivalente a $\exists x \neg \exists y (xy = 1)$, que é equivalente a $\exists x \forall y \neg (xy = 1)$. Como $\neg (xy = 1)$ pode ser expresso de maneira mais simples por $xy \neq 1$, concluímos que nossa sentença negada pode ser expressa como $\exists x \forall y (xy \neq 1)$. ◀

EXEMPLO 15 Use quantificadores para a sentença "Não existe uma mulher que já tenha tomado um avião em todas as linhas aéreas do mundo".

Solução: Essa sentença é a negação da sentença do Exemplo 13. De acordo com Exemplo 13, nossa sentença pode ser expressa por $\neg \exists w \forall a \exists f (P(w,f) \wedge Q(f,a))$, em que $P(w,f)$ é o predicado "w tomou o avião f" e $Q(f,a)$ é o predicado "f é um avião da linha a". Por sucessivas aplicações das leis de De Morgan para quantificadores na Tabela 2 da Seção 1.3, podemos mover a negação para dentro de todos os quantificadores e, por aplicação da lei de De Morgan para a conjunção no último passo, vemos que nossa sentença é equivalente a cada uma das outras desta seqüência de sentenças:

$$\begin{aligned} \forall w \neg \forall a \exists f (P(w,f) \wedge Q(f,a)) &\equiv \forall w \exists a \neg \exists f (P(w,f) \wedge Q(f,a)) \\ &\equiv \forall w \exists a \forall f \neg (P(w,f) \wedge Q(f,a)) \\ &\equiv \forall w \exists a \forall f (\neg P(w,f) \vee \neg Q(f,a)). \end{aligned}$$

Esta última sentença diz que "Para toda mulher existe uma linha aérea tal que, para todos os vôos, essa mulher não tomou esse vôo ou esse vôo não é dessa linha aérea". ◀

EXEMPLO 16 (*Requer cálculo*) Use quantificadores e predicados para expressar que $\lim_{x \to a} f(x)$ não existe.

Solução: Para dizer que $\lim_{x \to a} f(x)$ não existe, basta dizer que, para todo L, $\lim_{x \to a} f(x) \neq L$. Usando o Exemplo 8, a sentença $\lim_{x \to a} f(x) \neq L$ pode ser expressa por

$$\neg \forall \epsilon > 0 \; \exists \delta > 0 \; \forall x \; (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon).$$

Por sucessivas aplicações das regras para negação de expressões quantificadas, construímos esta seqüência de sentenças equivalentes

$$\begin{aligned}
&\neg\forall\,\epsilon > 0\ \exists\delta > 0\ \forall x(0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon)\\
&\quad\equiv \exists\epsilon > 0\ \neg\exists\delta > 0\ \forall x\,(0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon)\\
&\quad\equiv \exists\epsilon > 0\ \forall\delta > 0\ \neg\forall x\,(0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon)\\
&\quad\equiv \exists\epsilon > 0\ \forall\delta > 0\ \exists x\,\neg(0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon)\\
&\quad\equiv \exists\epsilon > 0\ \forall\delta > 0\ \exists x\,(0 < |x - a| < \delta \wedge |f(x) - L| \geq \epsilon).
\end{aligned}$$

Na última passagem, usamos a equivalência $\neg(p \rightarrow q) \equiv p \wedge \neg q$, que segue da quinta equivalência da Tabela 7 da Seção 1.2.

Como a sentença "$\lim_{x \to a} f(x)$ não existe" significa que para todo número real L, $\lim_{x \to a} f(x) \neq L$, essa pode ser expressa por

$$\forall L \exists\,\epsilon > 0\ \forall\delta > 0\ \exists x\,(0 < |x - a| < \delta \wedge |f(x) - L| \geq \epsilon).$$

Essa última sentença diz que para todo número real L existe um número real $\epsilon > 0$ tal que para cada número real $\delta > 0$, existe um número real x tal que $0 < |x - a| < \delta$ e $|f(x) - L| \geq \epsilon$. ◀

Exercícios

1. Transcreva as proposições abaixo para o português, em que o domínio para cada variável consista nos números reais.
- **a)** $\forall x \exists y(x < y)$
- **b)** $\forall x \forall y(((x \geq 0) \wedge (y \geq 0)) \rightarrow (xy \geq 0))$
- **c)** $\forall x \forall y \exists z(xy = z)$

2. Transcreva as proposições abaixo para o português, em que o domínio para cada variável consista nos números reais.
- **a)** $\exists x \forall y(xy = y)$
- **b)** $\forall x \forall y(((x \geq 0) \wedge (y < 0)) \rightarrow (x - y > 0))$
- **c)** $\forall x \forall y \exists z\,(x = y + z)$

3. Considere $Q(x, y)$ como a proposição "x enviou um e-mail para y", em que o domínio para x e y são todos os estudantes de sua sala. Expresse cada uma das quantificações abaixo em português.
- **a)** $\exists x \exists y Q(x, y)$ **b)** $\exists x \forall y Q(x, y)$
- **c)** $\forall x \exists y Q(x, y)$ **d)** $\exists y \forall x Q(x, y)$
- **e)** $\forall y \exists x Q(x, y)$ **f)** $\forall x \forall y Q(x, y)$

4. Considere $P(x, y)$ como a proposição "o estudante x tem freqüentado as aulas y", em que o domínio para x são os estudantes de sua sala e y, todos os cursos de ciência da computação em sua escola. Expresse cada uma das quantificações abaixo em português.
- **a)** $\exists x \exists y P(x, y)$ **b)** $\exists x \forall y P(x, y)$
- **c)** $\forall x \exists y P(x, y)$ **d)** $\exists y \forall x P(x, y)$
- **e)** $\forall y \exists x P(x, y)$ **f)** $\forall x \forall y P(x, y)$

5. Considere $W(x, y)$ que significa que o estudante x visitou o site da Web y, em que o domínio para x são todos os estudantes de sua escola e o domínio para y, todos os sites da Web. Expresse cada uma das proposições abaixo em uma sentença em português.
- **a)** W(Sarah Smith, www.att.com)
- **b)** $\exists x W(x,$ www.imdb.org)
- **c)** $\exists y W($Jose Orez, $y)$
- **d)** $\exists y(W($Ashok Puri, $y) \wedge W($Cindy Yoon, $y))$
- **e)** $\exists y \forall z\,(y \neq$ (David Belcher) $\wedge\ (W($David Belcher, $z) \rightarrow W(y,z)))$
- **f)** $\exists x \exists y \forall z((x \neq y) \wedge (W(x, z) \leftrightarrow W(y, z)))$

6. Considere $C(x, y)$, que significa que o estudante x está inscrito na aula y, em que o domínio para x são todos os estudantes de sua escola e o domínio para y, todas as aulas dadas na escola. Expresse cada uma das proposições abaixo em uma sentença em português.
- **a)** C(Randy Goldberg, CS 252)
- **b)** $\exists x C(x,$ Math 695)
- **c)** $\exists y C($Carol Sitea, $y)$
- **d)** $\exists x(C(x,$ Math 222) $\wedge\ C(x,$ CS 252))
- **e)** $\exists x \exists y \forall z(x \neq y) \wedge (C(x, z) \rightarrow C(y, z)))$
- **f)** $\exists x \exists y \forall z(x \neq y) \wedge (C(x, z) \leftrightarrow C(y, z)))$

7. Considere $T(x, y)$, que significa que o estudante x gosta da cozinha y, em que o domínio para x são todos os estudantes de sua escola e o domínio para y, todas as cozinhas. Expresse cada uma das proposições abaixo em uma sentença em português.
- **a)** $\neg T$(Abdallah Hussein, Japonesa)
- **b)** $\exists x T(x,$ Coreana) $\wedge\ \forall x T(x,$ Mexicana)

c) $\exists y T(\text{Monique Arsenault}, y) \vee T(\text{Jay Johnson}, y)$

d) $\forall x \forall z \exists y((x \neq z) \rightarrow \neg(T(x, y) \wedge T(z, y)))$

e) $\exists x \exists z \forall y(T(x, y) \leftrightarrow T(z, y))$

f) $\forall x \forall z \exists y(T(x, y) \leftrightarrow T(z, y))$

8. Considere $Q(x, y)$ como a proposição "o estudante x foi um participante no programa de perguntas e respostas y". Expresse cada uma das sentenças abaixo em termos de $Q(x, y)$, quantificadores e conectivos lógicos, em que o domínio para x são todos os estudantes de sua escola e o domínio de y, todos os programas de perguntas e respostas da televisão.

a) Há um estudante na sua escola que participou de um programa de perguntas e respostas na televisão.

b) Nenhum estudante da sua escola nunca participou de um programa de perguntas e respostas na televisão.

c) Há um estudante em sua escola que participou do programa *Show do Milhão* e *Roda da Fortuna*.

d) Todo programa de perguntas e resposta da televisão teve como participante um estudante de sua escola.

e) Pelo menos dois estudantes da sua escola foram participantes do *Show do Milhão*.

9. Considere $L(x, y)$ como a proposição "x ama y", em que o domínio para x e y são todas as pessoas do mundo. Use quantificadores para expressar cada proposição abaixo.

a) Todos amam Jerry.

b) Todas as pessoas amam alguém.

c) Há alguém que é amado por todos.

d) Ninguém ama a todos.

e) Há alguém a quem Lídia não ama.

f) Há alguém a quem ninguém ama.

g) Há exatamente uma pessoa a quem todos amam.

h) Há exatamente duas pessoas a quem Lynn ama.

i) Todos amam a si próprios.

j) Há alguém que não ama ninguém além dele próprio.

10. Considere $F(x, y)$ como a proposição "x pode enganar y", em que o domínio são todas as pessoas no mundo. Use quantificadores para expressar cada uma das proposições abaixo.

a) Todos podem enganar Fred.

b) Evelyn pode enganar a todos.

c) Todos podem enganar alguém.

d) Não há ninguém que possa enganar a todos.

e) Todos podem ser enganados por alguém.

f) Ninguém pode enganar Fred e Jerry.

g) Nancy pode enganar exatamente duas pessoas.

h) Há exatamente uma pessoa a quem todos podem enganar.

i) Ninguém pode enganar a si próprio.

j) Há alguém que pode enganar exatamente uma pessoa além de si próprio.

11. Considere $S(x)$ como o predicado "x é um estudante", $F(x)$ o predicado "x é um membro da faculdade" e $A(x, y)$ o predicado "x fez uma pergunta a y", em que o domínio são todas as pessoas associadas a sua escola. Use quantificadores para expressar cada proposição a seguir.

a) Lois fez uma pergunta ao professor Michaels.

b) Todo estudante fez uma pergunta ao professor Gross.

c) Todo membro da faculdade ou fez uma pergunta ao professor Miller ou foi questionado pelo professor Miller.

d) Algum estudante não fez nenhuma pergunta a qualquer membro da faculdade.

e) Há um membro da faculdade que nunca recebeu uma pergunta de um estudante.

f) Algum estudante fez uma pergunta a todos os membros da faculdade.

g) Há um membro da faculdade que fez uma pergunta a outro membro da faculdade.

h) Algum estudante nunca recebeu uma pergunta feita por um membro da faculdade.

12. Considere $I(x)$ como a proposição "x tem uma conexão de Internet" e $C(x, y)$ como a proposição "x e y conversaram pela Internet", em que o domínio para as variáveis x e y são todos os estudantes de sua sala. Use quantificadores para expressar cada uma das proposições abaixo.

a) Jerry não tem uma conexão de Internet.

b) Rachel não conversou pela Internet com Chelsea.

c) Jan e Sharon nunca conversaram pela Internet.

d) Ninguém na sala conversou pela Internet com Bob.

e) Sanjay conversou pela Internet com todos, menos com Joseph.

f) Alguém de sua sala não tem uma conexão de Internet.

g) Nem todos em sua sala têm uma conexão de Internet.

h) Exatamente um estudante em sua sala tem uma conexão de Internet.

i) Todos em sua sala, exceto um estudante, têm uma conexão de Internet.

j) Todos os estudantes na sua sala que têm conexão de Internet conversaram com pelo menos um outro estudante da sala.

k) Alguém de sua sala tem conexão de Internet, mas não conversou com ninguém mais de sua sala pela Internet.

l) Há dois estudantes em sua sala que não conversam entre si pela Internet.

m) Há um estudante em sua sala que conversa com todos da sala pela Internet.

n) Há pelo menos dois estudantes em sua sala que não conversam com a mesma pessoa pela Internet.

o) Há dois estudantes na sala que conversam apenas entre eles pela Internet.

13. Considere $M(x, y)$ como "x enviou um e-mail a y" e $T(x, y)$ como "x telefonou para y", em que o domínio são todos os estudantes em sua sala. Use quantificadores para expressar cada uma das proposições a seguir. (Assuma que todos os e-mails que foram enviados foram recebidos; o que geralmente não acontece.)

a) Chou nunca mandou um e-mail para Koko.

b) Arlene nunca mandou um e-mail ou telefonou para Sarah.

c) José nunca recebeu um e-mail de Débora.

d) Todo estudante de sua sala mandou um e-mail para Ken.

e) Ninguém da sua sala telefonou para Nina.

f) Alguém na sua sala telefonou ou mandou um e-mail para Avi.

g) Há um estudante na sua sala que enviou a mais alguém de sua sala um e-mail.

h) Há alguém na sua sala que ou mandou um e-mail ou telefonou a mais alguém de sua sala.

i) Há dois estudantes em sua sala que mandaram e-mail um para o outro.

j) Há um estudante que mandou para si próprio um e-mail.

k) Há um estudante na sua sala que não recebeu o e-mail que alguém da sala mandou e não recebeu telefonema de nenhum outro estudante na sala.

l) Todos os estudantes da sala ou receberam um e-mail ou receberam um telefonema de outro estudante da sala.

m) Há pelos menos dois estudantes de sua sala, dos quais o primeiro mandou um e-mail para o segundo e este telefonou para aquele.

n) Há dois estudantes diferentes que mandaram e-mails entre si ou telefonaram para outra pessoa da sala.

14. Use quantificadores e predicados com mais de uma variável para expressar as proposições abaixo.

a) Há um estudante nesta sala que fala hindu.

b) Todo estudante desta sala pratica algum esporte.

c) Algum estudante nesta sala visitou o Alasca, mas não visitou o Havaí.

d) Todos os estudantes desta sala aprenderam pelo menos uma linguagem computacional.

e) Há um estudante nesta sala que cursou todas as disciplinas oferecidas por um dos departamentos da escola.

f) Algum estudante desta sala cresceu na mesma cidade que outro estudante da sala.

g) Todo estudante desta sala conversou pela Internet com pelo menos outro estudante, em pelo menos um grupo de bate-papo.

15. Use quantificadores e predicados com mais de uma variável para expressar as proposições abaixo.

a) Todo estudante de ciência da computação precisa de um curso de matemática discreta.

b) Há um estudante nesta sala que possui seu próprio computador.

c) Todo estudante nesta sala participou de pelo menos um curso de ciência da computação.

d) Há um estudante nesta sala que participou de pelo menos um curso de ciência da computação.

e) Todo estudante desta sala já esteve em todos os prédios do campus.

f) Há um estudante desta sala que já esteve em todas as salas de pelo menos um prédio do campus.

g) Todo estudante nesta sala já esteve em pelo menos uma sala de todos os prédios do campus.

16. Uma aula de matemática discreta contém um graduando em matemática que é um calouro, 12 matemáticos que são veteranos, 15 cientistas da computação que são veteranos, 2 matemáticos que são juniores, 2 cientistas da computação que são juniores e 1 cientista da computação que é sênior. Expresse cada uma das proposições abaixo em termos de quantificadores e depois determine seu valor-verdade.

a) Há um estudante na sala que é júnior.

b) Todo estudante na sala é graduado em ciência da computação.

c) Há um estudante na sala que não é nem graduando em matemática nem júnior.

d) Todo estudante na sala é ou um veterano ou um graduando em ciência da computação.

e) Há um graduando, tal que existe um outro estudante na sala que em todo ano estudou com este graduando.

17. Expresse cada um dos sistemas de especificações abaixo usando predicados, quantificadores e conectivos lógicos, se necessário.

a) Todo usuário tem acesso a exatamente uma caixa de entrada.

b) Há um processo que continua executando durante todas as condições de erro se o kernel estiver funcionando corretamente.

c) Todos os usuários do campus têm acesso a todos os sites da Web de extensão .edu.

***d)** Há exatamente dois sistemas que monitoram todos os servidores remotos.

18. Expresse cada um dos sistemas de especificações abaixo usando predicados, quantificadores e conectivos lógicos, se necessário.

a) Pelo menos um console deve estar acessível durante toda condição de cuidado.

b) O endereço de e-mail de todo usuário deve ser recuperado assim que o arquivo contiver pelo menos uma mensagem enviada pelo usuário no sistema.

c) Para toda quebra de segurança há pelo menos um mecanismo que pode apagar aquela quebra se, e apenas se, um processo não for comprometido.

d) Há pelo menos dois caminhos conectados a dois pontos distintos na rede de transmissão.

e) Ninguém sabe a senha de todos os usuários do sistema, exceto o administrador do sistema, que sabe todas as senhas.

19. Expresse cada uma das proposições abaixo usando operadores matemáticos e lógicos, predicados e quantificadores, em que o domínio são todos os números inteiros.

a) A soma de dois números inteiros negativos é negativa.

b) A diferença de dois números inteiros positivos não é necessariamente positiva.

c) A soma dos quadrados de dois números inteiros é maior que ou igual ao quadrado de sua soma.

d) O valor absoluto do produto de dois números inteiros é o produto de seus valores absolutos.

20. Expresse cada uma das proposições a seguir usando predicados, quantificadores, conectivos lógicos e operadores matemáticos, em que o domínio são todos os números inteiros.

a) O produto de dois números inteiros negativos é positivo.

b) A média de dois números inteiros positivos é positiva.

c) A diferença entre dois números inteiros negativos não é necessariamente negativa.
d) O valor absoluto da soma de dois números inteiros não excede à soma dos valores absolutos desses números inteiros.

21. Use predicados, quantificadores, conectivos lógicos e operadores matemáticos para expressar a proposição de que todo número inteiro positivo é a soma dos quadrados de quatro números inteiros.

22. Use predicados, quantificadores, conectivos lógicos e operadores matemáticos para expressar a proposição de que há um número inteiro positivo que não é a soma de três quadrados.

23. Expresse cada uma das proposições matemáticas abaixo usando predicados, quantificadores, conectivos lógicos e operadores matemáticos.
a) O produto de dois números reais negativos é positivo.
b) A diferença entre um número real e ele mesmo é zero.
c) Todo número real positivo tem exatamente duas raízes quadradas.
d) Um número real negativo não tem uma raiz quadrada que seja um número real.

24. Transcreva cada uma das quantificações agrupadas em proposições em português que expressem um fato matemático. O domínio em cada caso são todos os números reais.
a) $\exists x \forall y (x + y = y)$
b) $\forall x \forall y (((x \geq 0) \wedge (y < 0)) \rightarrow (x - y > 0))$
c) $\exists x \exists y (((x \leq 0) \wedge (y \leq 0)) \wedge (x - y > 0))$
d) $\forall x \forall y ((x \neq 0) \wedge (y \neq 0) \leftrightarrow (xy \neq 0))$

25. Transcreva cada uma das quantificações agrupadas em proposições em português que expressem um fato matemático. O domínio em cada caso são todos os números reais.
a) $\exists x \forall y (xy = y)$
b) $\forall x \forall y (((x < 0) \wedge (y < 0)) \rightarrow (xy > 0))$
c) $\exists x \exists y (((x^2 > y) \wedge (x < y))$
d) $\forall x \forall y \exists z (x + y = z)$

26. Considere $Q(x, y)$ como a proposição "$x + y = x - y$". Se o domínio das duas variáveis forem todos os números inteiros, quais são os valores-verdade?
a) $Q(1, 1)$ **b)** $Q(2, 0)$
c) $\forall y Q(1, y)$ **d)** $\exists x Q(x, 2)$
e) $\exists x \exists y Q(x, y)$ **f)** $\forall x \exists y Q(x, y)$
g) $\exists y \forall x Q(x, y)$ **h)** $\forall y \exists x Q(x, y)$
i) $\forall x \forall y Q(x, y)$

27. Determine o valor-verdade de cada uma das proposições abaixo se o domínio para as variáveis são todos os números inteiros.
a) $\forall n \exists m (n^2 < m)$ **b)** $\exists n \forall m (n < m^2)$
c) $\forall n \exists m (n + m = 0)$ **d)** $\exists n \forall m (nm = m)$
e) $\exists n \exists m (n^2 + m^2 = 5)$ **f)** $\exists n \exists m (n^2 + m^2 = 6)$
g) $\exists n \exists m (n + m = 4 \wedge n - m = 1)$
h) $\exists n \exists m (n + m = 4 \wedge n - m = 2)$
i) $\forall n \forall m \exists p (p = (m + n)/2)$

28. Determine o valor-verdade de cada uma das proposições a seguir se o domínio para as variáveis são todos os números reais.
a) $\forall x \exists y (x^2 = y)$ **b)** $\forall x \exists y (x = y^2)$
c) $\exists x \forall y (xy = 0)$
d) $\exists x \exists y (x + y \neq y + x)$
e) $\forall x (x \neq 0 \rightarrow \exists y (xy = 1))$
f) $\exists x \forall y (y \neq 0 \rightarrow xy = 1)$
g) $\forall x \exists y (x + y = 1)$
h) $\exists x \exists y (x + 2y = 2 \wedge 2x + 4y = 5)$
i) $\forall x \exists y (x + y = 2 \wedge 2x - y = 1)$
j) $\forall x \forall y \exists z (z = (x + y)/2)$

29. Suponha que o domínio da função proposicional $P(x, y)$ são os pares x e y, em que x é 1, 2 ou 3 e y, 1, 2 ou 3. Desenvolva as proposições abaixo usando disjunções e conjunções.
a) $\forall x \forall y P(x, y)$ **b)** $\exists x \exists y P(x, y)$
c) $\exists x \forall y P(x, y)$ **d)** $\forall y \exists x P(x, y)$

30. Reescreva cada uma das proposições para que as negações apareçam apenas inseridas nos predicados (ou seja, nenhuma negação esteja do lado de fora de um quantificador ou de uma expressão que envolva conectivos lógicos).
a) $\neg \exists y \exists x P(x, y)$ **b)** $\neg \forall x \exists y P(x, y)$
c) $\neg \exists y (Q(y) \wedge \forall x \neg R(x, y))$
d) $\neg \exists y (\exists x R(x, y) \vee \forall x S(x, y))$
e) $\neg \exists y (\forall x \exists z T(x, y, z) \vee \exists x \forall z U(x, y, z))$

31. Expresse as negações de cada uma das proposições abaixo, tal que todos os símbolos de negação precedam imediatamente os predicados.
a) $\forall x \exists y \forall z T(x, y, z)$
b) $\forall x \exists y P(x, y) \vee \forall x \exists y Q(x, y)$
c) $\forall x \exists y (P(x, y) \wedge \exists z R(x, y, z))$
d) $\forall x \exists y (P(x, y) \rightarrow Q(x, y))$

32. Expresse as negações de cada uma das proposições abaixo, tal que todos os símbolos de negação precedam imediatamente os predicados.
a) $\exists z \forall y \forall x T(x, y, z)$
b) $\exists x \exists y P(x, y) \wedge \forall x \forall y Q(x, y)$
c) $\exists x \exists y (Q(x, y) \leftrightarrow Q(y, x))$
d) $\forall y \exists x \exists z (T(x, y, z) \vee Q(x, y))$

33. Reescreva cada uma das proposições para que as negações apareçam apenas inseridas nos predicados (ou seja, nenhuma negação esteja do lado de fora de um quantificador ou de uma expressão que envolva conectivos lógicos).
a) $\neg \forall x \forall y P(x, y)$ **b)** $\neg \forall y \exists x P(x, y)$
c) $\neg \forall y \forall x (P(x, y) \vee Q(x, y))$
d) $\neg (\exists x \exists y \neg P(x, y) \wedge \forall x \forall y Q(x, y))$
e) $\neg \forall x (\exists y \forall z P(x, y, z) \wedge \exists z \forall y P(x, y, z))$

34. Encontre um domínio comum para as variáveis x, y e z para que a proposição $\forall x \forall y ((x \neq y) \rightarrow \forall z ((z = x) \vee (z = y)))$ seja verdadeira e outro domínio para que ela seja falsa.

35. Encontre um domínio comum para as variáveis x, y, z e w para que a proposição $\forall x \forall y \forall z \exists w ((w \neq x) \wedge (w \neq y) \wedge (w \neq z))$ seja verdadeira e outro domínio para que ela seja falsa.

36. Expresse cada uma das proposições a seguir. Então, forme a negação de cada proposição, para que a negação fique do lado esquerdo de um quantificador. Depois, expresse a negação em português. (Não use as palavras "Não é o caso de".)

a) Ninguém perdeu mais de mil dólares apostando na loteria.

b) Há um estudante na sala que conversou pela Internet com apenas um outro estudante.

c) Nenhum estudante nesta sala enviou um e-mail para dois outros estudantes da sala.

d) Algum estudante resolveu todos os exercícios deste livro.

e) Nenhum estudante resolveu pelo menos um exercício em todas as seções deste livro.

37. Expresse cada uma das proposições abaixo. Então, forme a negação de cada proposição, para que a negação fique do lado esquerdo de um quantificador. Depois, expresse a negação em português. (Não use as palavras "Não é o caso de".)

a) Todo estudante na sala freqüentou exatamente duas aulas de matemática nesta escola.

b) Alguém visitou todos os países do mundo, exceto a Líbia.

c) Ninguém escalou todas as montanhas do Himalaia.

d) Todo ator de cinema já participou de um filme ao lado de Kevin Bacon ou contracenou com alguém que já filmou com Kevin Bacon.

38. Expresse a negação das proposições abaixo usando quantificadores, e em português.

a) Todo estudante desta sala gosta de matemática.

b) Há um estudante nesta sala que nunca viu um computador.

c) Há um estudante nesta sala que freqüentou todos os cursos de matemática oferecidos nesta escola.

d) Há um estudante nesta sala que esteve em pelo menos uma sala de todos os prédios do campus.

39. Encontre um contra-exemplo, se possível, para estas proposições de quantificadores universais, em que o domínio para todas as variáveis são todos os números inteiros.

a) $\forall x \forall y (x^2 = y^2 \rightarrow x = y)$

b) $\forall x \exists y (y^2 = x)$

c) $\forall x \forall y (xy \geq x)$

40. Encontre um contra-exemplo, se possível, para estas proposições de quantificadores universais, em que o domínio para todas as variáveis são todos os números inteiros.

a) $\forall x \exists y (x = 1/y)$

b) $\forall x \exists y (y^2 - x < 100)$

c) $\forall x \forall y (x^2 \neq y^3)$

41. Use quantificadores para expressar a propriedade associativa para a multiplicação de números reais.

42. Use quantificadores para expressar as propriedades distributivas para a multiplicação sobre a adição de números reais.

43. Use quantificadores e conectivos lógicos para expressar o fato de que todo polinômio linear (ou seja, o polinômio de grau 1) com coeficientes reais e no qual o coeficiente de x é diferente de zero tem exatamente uma raiz real.

44. Use quantificadores e conectivos lógicos para expressar o fato de que um polinômio quadrático com coeficientes de números reais tem no máximo duas raízes reais.

45. Determine o valor-verdade da proposição $\forall x \exists y (xy = 1)$, se o domínio para as variáveis forem:

a) os números reais diferentes de zero.

b) os números inteiros diferentes de zero.

c) os números reais positivos.

46. Determine o valor-verdade da proposição $\exists x \forall y (x \leq y^2)$, se o domínio para as variáveis forem:

a) os números reais positivos.

b) os números inteiros.

c) os números reais diferentes de zero.

47. Mostre que as duas proposições $\neg \exists x \forall y P(x, y)$ e $\forall x \exists y \neg P(x, y)$, em que os dois quantificadores da primeira variável em $P(x, y)$ têm o mesmo domínio, e os dois quantificadores da segunda variável em $P(x, y)$ têm o mesmo domínio, são equivalentes logicamente.

***48.** Mostre que $\forall x P(x) \vee \forall x Q(x)$ e $\forall x \forall y P(x) \vee Q(y))$, em que todos os quantificadores têm o mesmo domínio não vazio, são equivalentes logicamente. (A nova variável y é utilizada para combinar corretamente as quantificações.)

***49.** **a)** Mostre que $\forall x P(x) \wedge \exists x Q(x)$ é equivalente logicamente a $\forall x \exists y (P(x) \wedge Q(y))$, em que todos os quantificadores têm o mesmo domínio não vazio.

b) Mostre que $\forall x P(x) \vee \exists x Q(x)$ é equivalente a $\forall x \exists y (P(x) \vee Q(y))$, em que todos os quantificadores têm o mesmo domínio não vazio.

Uma proposição está em **forma normal prenex (FNP)** se e apenas se estiver na forma

$$Q_1 x_1 Q_2 x_2 \ldots Q_k x_k P(x_1, x_2, \ldots, x_k),$$

em que cada Q_i, $i = 1, 2, \ldots, k$, é ou um quantificador de existência ou um quantificador universal e $P(x_1, \ldots, x_k)$ é um predicado que não envolva quantificadores. Por exemplo, $\exists x \forall y (P(x, y) \wedge Q(y))$ está em forma normal prenex, enquanto $\exists x P(x) \vee \forall x Q(x)$ não está (porque os quantificadores não aparecem todos primeiro).

Toda proposição formada de variáveis proposicionais, predicados, **V** e **F** usando conectivos lógicos e quantificadores é equivalente a uma proposição em forma normal prenex. O Exercício 51 pede uma demonstração desse fato.

***50.** Coloque as proposições abaixo em forma normal prenex. [*Dica:* Use equivalentes lógicos das tabelas 6 e 7 da Seção 1.2, tabela 2 da Seção 1.3, Exemplo 19 da Seção 1.3, exercícios 45 e 46 da Seção 1.3 e exercícios 48 e 49 desta seção.]

a) $\exists x P(x) \vee \exists x Q(x) \vee A$, em que A é uma proposição que não envolve nenhum quantificador.

b) $\neg (\forall x P(x) \vee \forall x Q(x))$

c) $\exists x P(x) \rightarrow \exists x Q(x)$

****51.** Mostre como transformar uma proposição arbitrária em uma proposição em forma normal prenex equivalente à proposição dada.

***52.** Expresse a quantificação $\exists! x P(x)$, introduzida na página 37, usando quantificações universais, quantificações de existência e operadores lógicos.

1.5 Regras de Inferência

Introdução

Mais adiante, neste capítulo, vamos estudar demonstrações. Demonstrações em matemática são argumentos válidos que estabelecem a veracidade das sentenças matemáticas. Por um **argumento**, entendemos uma seqüência de sentenças que terminam com uma conclusão e, por **válido**, que uma conclusão, ou a sentença final do argumento, deve seguir o valor-verdade das sentenças precedentes, ou **premissas**, do argumento. Ou seja, um argumento é válido se e somente se for impossível que todas as premissas sejam verdadeiras e a conclusão seja falsa. Para deduzir novas sentenças de sentenças que já temos, usamos regras de inferência, as quais são moldes para construção de argumentos válidos. Regras de inferência são nossas ferramentas básicas para o estabelecimento do valor-verdade das sentenças.

Antes de estudarmos demonstrações matemáticas, vamos olhar para argumentos que envolvem apenas proposições compostas. Vamos definir o que significa um argumento ser válido quando envolve proposições compostas. Então, vamos introduzir um conjunto de regras de inferência de lógica proposicional. Essas regras de inferência são o mais importante ingrediente na produção de argumentos válidos. Depois de ilustrar como as regras de inferência são utilizadas para produzir argumentos válidos, vamos descrever algumas formas comuns de raciocínio incorreto, chamadas de **falácias**, que nos levam a argumentos inválidos.

Depois de estudar as regras de inferência em lógica proposicional, vamos introduzir regras de inferência para sentenças quantificadas. Vamos descrever como essas regras de inferência podem ser utilizadas para produzir argumentos válidos. Essas regras de inferência para sentenças que envolvem quantificadores universal e existencial representam importante papel em demonstrações em ciência da computação e matemática, embora elas sejam sempre utilizadas sem serem explicitamente mencionadas.

Finalmente, vamos mostrar como regras de inferência para sentenças proposicionais e quantificacionais podem ser combinadas. Essas combinações são freqüentemente utilizadas em argumentos complicados.

Argumentos Válidos em Lógica Proposicional

Considere o seguinte argumento que envolve proposições (o qual, por definição, é uma seqüência de proposições):

"Se você tem uma senha atualizada, então você pode entrar na rede."
"Você tem uma senha atualizada."

Portanto,

"Você pode entrar na rede."

Gostaríamos de determinar quando este argumento é válido. Ou seja, gostaríamos de determinar se a conclusão "Você pode entrar na rede" deve ser verdadeira quando as premissas "Se você tem uma senha atualizada, então você pode entrar na rede" e "Você tem uma senha atualizada" também forem ambas verdadeiras.

Antes de discutir a validade deste argumento particular, vamos olhar para sua forma. Use p para representar "Você tem uma senha atualizada" e q para representar "Você pode entrar na rede". Então, o argumento tem a forma

$$\begin{array}{l} p \rightarrow q \\ p \\ \hline \therefore q \end{array}$$

em que $\therefore$ é o símbolo para indicar "portanto".

Sabemos que se p e q são variáveis proposicionais, a sentença $((p \rightarrow q) \wedge p) \rightarrow q$ é uma tautologia (veja o Exercício 10(c) na Seção 1.2). Em particular, quando ambos $p \rightarrow q$ e p são verdadeiras, sabemos que q também deve ser. Dizemos que essa é uma forma **válida** de argumento porque sempre que todas as suas premissas (todas as sentenças do argumento, a não ser a última, a conclusão) são verdadeiras, a conclusão também deve ser. Agora suponha que ambas "Se você tem uma senha atualizada, então você pode entrar na rede" e "Você tem uma senha atualizada" são sentenças verdadeiras. Quando trocamos p por "Você tem uma senha atualizada" e q por "Você pode entrar na rede", segue necessariamente que a conclusão "Você pode entrar na rede" é verdadeira. Esse argumento é **válido** porque está na forma válida. Note que sempre que substituirmos p e q por proposições em que $p \rightarrow q$ e p são verdadeiras, então q deve ser verdadeira.

O que acontece quando substituímos p e q nessa forma de argumento por proposições tal que p e $p \rightarrow q$ não são ambas verdadeiras? Por exemplo, suponha que p represente "Você tem acesso à rede" e q represente "Você pode mudar suas notas" e p seja verdadeira, mas $p \rightarrow q$ seja falsa. O argumento que obtemos substituindo esses valores de p e q na forma do argumento anterior é:

"Se você tem acesso à rede, então você pode mudar suas notas."
"Você tem acesso à rede."
$\therefore$ "Você pode mudar suas notas."

O argumento obtido é um argumento válido, mas, como uma das premissas, chamada de primeira premissa, é falsa, não podemos decidir se a conclusão é verdadeira. (Mas parece que essa conclusão é falsa.)

Em nossa discussão, para analisar um argumento, substituímos as proposições por variáveis proposicionais. Isso transforma um argumento em uma **forma de argumento**. Dizemos que a validade de um argumento segue a validade da forma do argumento. Resumimos a terminologia utilizada para discutir a validade de argumentos com nossa definição dessas noções importantes.

DEFINIÇÃO 1

Um *argumento* em lógica proposicional é uma seqüência de proposições. Todas, menos a última das proposições, são chamadas de *premissas,* e a última é chamada de *conclusão*. Um argumento é *válido* se a veracidade das premissas implica que a conclusão seja verdadeira.

Uma *forma de argumento* em lógica proposicional é a seqüência de proposições compostas que envolvem variáveis proposicionais. Uma forma de argumento é *válida* quaisquer que sejam as proposições substituídas nas variáveis proposicionais em suas sentenças; a conclusão é verdadeira se as premissas forem todas verdadeiras.

Da definição de forma de argumento válida, vemos que uma forma de argumento com premissas $p_1, p_2, p_3, \ldots, p_n$ e conclusão q é válida, quando $(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \rightarrow q$ é uma tautologia.

A chave para mostrar que um argumento na lógica proposicional é válido é mostrar que sua forma de argumento é válida. Conseqüentemente, gostaríamos de ter técnicas para mostrar que formas de argumentos são válidas. Vamos agora desenvolver métodos para alcançar esse objetivo.

Regras de Inferência para Lógica Proposicional

Podemos sempre usar uma tabela-verdade para mostrar que uma forma de argumento é válida. Fazemos isso mostrando que sempre que as premissas são verdadeiras, a conclusão deve ser verdadeira também. No entanto, esse pode ser um modo um pouco tedioso. Por exemplo, quando uma forma de argumento envolve 10 variáveis proposicionais diferentes, usar uma tabela-verdade para mostrar que esse argumento é válido requer $2^{10} = 1.024$ linhas diferentes. Felizmente, não

precisamos sempre recorrer às tabelas-verdade. Em vez disso, podemos estabelecer a validade de algumas formas de argumento relativamente simples, chamadas de **regras de inferência**. Essas regras de inferência podem ser utilizadas como tijolos para construir formas de argumento válidas mais complicadas. Vamos agora introduzir a mais importante das regras de inferência na lógica proposicional.

A tautologia $(p \wedge (p \rightarrow q)) \rightarrow q$ é a base da regra de inferência chamada de *modus ponens*, ou **propriedade de destacamento**. (*Modus ponens*, em latim, significa *modo que afirma.*) Essa tautologia nos leva à seguinte forma de argumento válida, que já foi vista na nossa discussão sobre argumentos (em que, como antes, o $\therefore$ é o símbolo para indicar "portanto"):

$$\begin{array}{l} p \\ p \rightarrow q \\ \hline \therefore q \end{array}$$

Usando essa notação, as hipóteses são escritas em colunas, seguidas por uma barra horizontal, seguida por uma linha que começa com o símbolo "portanto" e termina com a conclusão. Em particular, *modus ponens* nos diz que se uma sentença condicional e a hipótese dessa sentença condicional são verdadeiras, então a conclusão também deve ser verdadeira. O Exemplo 1 ilustra o uso do *modus ponens*.

EXEMPLO 1 Suponha que a sentença condicional "Se nevar hoje, então eu vou esquiar" e sua hipótese "Está nevando hoje" são verdadeiras. Então, por *modus ponens*, segue que a conclusão do condicional "Vou esquiar" é verdadeira. ◀

Como mencionado anteriormente, um argumento válido pode nos levar a uma conclusão incorreta se uma ou mais de suas premissas são falsas. Ilustramos isso, novamente, no Exemplo 2.

EXEMPLO 2 Determine se o argumento dado aqui é válido e se sua conclusão deve ser verdadeira apenas pela validade do argumento.

"Se $\sqrt{2} > \frac{3}{2}$, então $(\sqrt{2})^2 > (\frac{3}{2})^2$. Sabemos que $\sqrt{2} > \frac{3}{2}$. Conseqüentemente, $(\sqrt{2})^2 = 2 > (\frac{3}{2})^2 = \frac{9}{4}$."

Solução: Seja p a proposição "$\sqrt{2} > \frac{3}{2}$" e q a proposição "$2 > (\frac{3}{2})^2$". As premissas do argumento são $p \rightarrow q$ e p, e q é a conclusão. Esse argumento é válido, pois é construído de acordo com *modus ponens*, uma forma válida de argumento. No entanto, uma de suas premissas, $\sqrt{2} > \frac{3}{2}$, é falsa. Conseqüentemente, não podemos deduzir que a conclusão seja verdadeira. Nesse caso, notamos que a conclusão é falsa, pois $2 < \frac{9}{4}$. ◀

A Tabela 1 lista as mais importantes regras de inferência para a lógica proposicional. Os exercícios 9, 10, 15 e 30 na Seção 1.2 pedem verificações de que essas regras de inferência são formas válidas de argumento. Agora daremos exemplos de argumentos que usam essas regras de inferência. Em cada argumento, primeiro usaremos variáveis proposicionais para expressar as proposições no argumento. Depois, mostraremos que a forma resultante de argumento é uma regra da Tabela 1.

EXEMPLO 3 Diga qual regra de inferência é a base do seguinte argumento: "Está esfriando agora. Portanto, está esfriando ou chovendo agora".

TABELA 1 Regras de Inferência.

Regra de Inferência	*Tautologia*	*Nome*
p $p \rightarrow q$ $\therefore q$	$[p \wedge (p \rightarrow q)] \rightarrow q$	*Modus ponens*
$\neg q$ $p \rightarrow q$ $\therefore \neg p$	$[\neg q \wedge (p \rightarrow q)] \rightarrow \neg p$	*Modus tollens*
$p \rightarrow q$ $q \rightarrow r$ $\therefore p \rightarrow r$	$[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$	Silogismo hipotético
$p \vee q$ $\neg p$ $\therefore q$	$[(p \vee q) \wedge \neg p] \rightarrow q$	Silogismo disjuntivo
p $\therefore p \vee q$	$p \rightarrow (p \vee q)$	Adição
$p \wedge q$ $\therefore p$	$(p \wedge q) \rightarrow p$	Simplificação
p q $\therefore p \wedge q$	$[(p) \wedge (q)] \rightarrow (p \wedge q)$	Conjunção
$p \vee q$ $\neg p \vee r$ $\therefore q \vee r$	$[(p \vee q) \wedge (\neg p \vee r)] \rightarrow (q \vee r)$	Resolução

Solução: Seja p a proposição "Está esfriando agora" e q a proposição "Está chovendo agora". Então esse argumento é da forma

$$\begin{array}{l} p \\ \hline \therefore p \vee q \end{array}$$

Esse é um argumento que usa a regra da adição. ◀

EXEMPLO 4 Diga qual regra de inferência é a base do seguinte argumento: "Está esfriando e chovendo agora. Portanto, está esfriando agora".

Solução: Seja p a proposição "Está esfriando agora" e q a proposição "Está chovendo agora". Então, esse argumento é da forma

$$\begin{array}{l} p \wedge q \\ \hline \therefore p \end{array}$$

Esse é um argumento que usa a regra da simplificação. ◀

EXEMPLO 5 Diga qual regra de inferência é a base do seguinte argumento:

> "Se chover, então não haverá churrasco hoje. Se não houver churrasco hoje, haverá amanhã. Portanto, se chover hoje, então haverá churrasco amanhã."

Solução: Seja p a proposição "Está chovendo hoje", q a proposição "Não terá churrasco hoje" e r a proposição "Terá churrasco amanhã". Então esse argumento é da forma

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$

Logo, esse argumento é um silogismo hipotético. ◀

Usando Regras de Inferência para Construir Argumentos

Quando existem muitas premissas, muitas regras de inferência são freqüentemente necessárias para mostrar que um argumento é válido. Isso é ilustrado nos exemplos 6 e 7, onde os passos do argumento são dispostos em linhas separadas, com a razão para cada passo explicitamente colocada ao lado. Esses exemplos também mostram como argumentos em português podem ser analisados por meio de regras de inferência.

EXEMPLO 6 Mostre que as hipóteses "Não está ensolarada esta tarde e está mais frio que ontem", "Vamos nadar se estiver ensolarado", "Se não formos nadar, então vamos fazer um passeio de barco" e "Se fizermos um passeio de barco, então estaremos em casa ao anoitecer" nos levam à conclusão "Estaremos em casa ao anoitecer".

Exemplos Extras

Solução: Seja p a proposição "Está ensolarada esta tarde", q a proposição "Está mais frio que ontem", r a proposição "Vamos nadar", s a proposição "Vamos fazer um passeio de barco" e t a proposição "Estaremos em casa ao anoitecer". Então, as hipóteses se tornam $\neg p \wedge q$, $r \rightarrow p$, $\neg r \rightarrow s$, e $s \rightarrow t$ e a conclusão é simplesmente t.

Construímos um argumento para mostrar que nossas hipóteses nos levam à conclusão como se segue.

Passo	**Razão**
1. $\neg p \wedge q$	Hipótese
2. $\neg p$	Simplificação usando (1)
3. $r \rightarrow p$	Hipótese
4. $\neg r$	*Modus tollens* usando (2) e (3)
5. $\neg r \rightarrow s$	Hipótese
6. s	*Modus ponens* usando (4) e (5)
7. $s \rightarrow t$	Hipótese
8. t	*Modus ponens* usando (6) e (7)

Note que poderíamos ter empregado uma tabela-verdade para mostrar que, sempre que as hipóteses são verdadeiras, a conclusão também será. No entanto, como estamos trabalhando com cinco variáveis proposicionais (p, q, r, s e t), essa tabela teria 32 linhas. ◀

EXEMPLO 7 Mostre que as hipóteses "Se você me mandar um e-mail, então eu terminarei o programa", "Se você não me mandar um e-mail, então vou dormir cedo" e "Se eu dormir cedo, então

acordarei sentindo-me bem" nos levam à conclusão "Se eu não terminar o programa, então eu acordarei sentindo-me bem".

Solução: Seja p a proposição "Você me manda um e-mail", q a proposição "Eu terminarei o programa", r a proposição "Eu vou dormir cedo" e s a proposição "Eu acordarei sentindo-me bem". Então as hipóteses são $p \rightarrow q$, $\neg p \rightarrow r$ e $r \rightarrow s$. A conclusão desejada é $\neg q \rightarrow s$. Temos de dar um argumento válido com as hipóteses $p \rightarrow q$, $\neg p \rightarrow r$ e $r \rightarrow s$ e a conclusão $\neg q \rightarrow s$.

Essa forma de argumento mostra que as hipóteses nos levam à conclusão desejada.

Passo	Razão
1. $p \rightarrow q$	Hipótese
2. $\neg q \rightarrow \neg p$	Contrapositiva de (1)
3. $\neg p \rightarrow r$	Hipótese
4. $\neg q \rightarrow r$	Silogismo hipotético usando (2) e (3)
5. $r \rightarrow s$	Hipótese
6. $\neg q \rightarrow s$	Silogismo hipotético usando (4) e (5)

◀

Resolução

Programas de computador têm sido desenvolvidos para automatizar a tarefa de raciocinar e fornecer teoremas. Muitos desses programas fazem uso da regra de inferência conhecida como **resolução**. Essa regra de inferência baseia-se na tautologia

Links

$$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r).$$

(A verificação de que essa sentença é uma tautologia foi pedida no Exercício 30 na Seção 1.2.) A disjunção final da regra, $q \vee r$, é chamada de **resolvente**. Quando $q = r$ na tautologia, obtemos $(p \vee q) \wedge (\neg p \vee q) \rightarrow q$. Além disso, quando $r = \mathbf{F}$, obtemos $(p \vee q) \wedge (\neg p) \rightarrow q$ (porque $q \vee \mathbf{F} \equiv q$), que é a tautologia na qual está fundamentada a regra do silogismo disjuntivo.

EXEMPLO 8 Use a resolução para mostrar que as hipóteses "Jasmim está esquiando ou não está nevando" e "Está nevando ou José está jogando futebol" implica que "Jasmim está esquiando ou José está jogando futebol".

Exemplos Extras

Solução: Seja p a proposição "Está nevando", q a proposição "Jasmim está esquiando", e r a proposição "José está jogando futebol". Podemos representar as hipóteses por $\neg p \vee q$ e $p \vee r$, respectivamente. Usando resolução, a proposição $q \vee r$, "Jasmim está esquiando ou José está jogando futebol", pode ser concluída. ◀

A resolução tem um importante papel em linguagens de programação fundamentadas nas regras da lógica, como Prolog (onde as regras de resolução para sentenças quantificadas são aplicadas). Além disso, pode ser utilizada para construir sistemas que provêm teoremas automaticamente. Para construir demonstrações em lógica proposicional usando resolução como a única regra de inferência, as hipóteses e a conclusão devem ser expressas como **cláusulas**, em que uma cláusula é uma disjunção de variáveis ou das negações dessas variáveis. Podemos substituir uma sentença em lógica proposicional que não é uma cláusula por uma ou mais sentenças equivalentes que são cláusulas. Por exemplo, suponha que tenhamos uma sentença da forma $p \vee (q \wedge r)$. Como $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$, podemos substituir a sentença $p \vee (q \wedge r)$ por duas sentenças

$p \vee q$ e $p \vee r$, cada uma delas sendo uma cláusula. Podemos substituir uma sentença da forma $\neg(p \vee q)$ por duas sentenças $\neg p$ e $\neg q$ porque a lei de De Morgan nos diz que $\neg(p \vee q) \equiv \neg p \wedge \neg q$. Podemos também substituir uma sentença condicional $p \rightarrow q$ pela disjunção equivalente $\neg p \vee q$.

EXEMPLO 9 Mostre que as hipóteses $(p \wedge q) \vee r$ e $r \rightarrow s$ implicam a conclusão $p \vee s$.

Solução: Podemos reescrever a hipótese $(p \wedge q) \vee r$ como duas cláusulas, $p \vee r$ e $q \vee r$. Podemos também substituir $r \rightarrow s$ pela cláusula equivalente $\neg r \vee s$. Usando as duas cláusulas $p \vee r$ e $\neg r \vee s$, podemos usar uma resolução para concluir $p \vee r$. ◀

Falácias

Muitas falácias comuns aparecem em argumentos incorretos. Essas falácias assemelham-se a regras de inferência, mas baseiam-se em contingências em vez de tautologias. Isso será discutido aqui para mostrar a distinção entre um raciocínio correto e um incorreto.

Links A proposição $[(p \rightarrow q) \wedge q] \rightarrow p$ não é uma tautologia, pois é falsa quando p é falsa e q é verdadeira. No entanto, existem muitos argumentos incorretos que tratam isso como uma tautologia. Em outras palavras, eles tratam o argumento com premissas $p \rightarrow q$ e q e conclusão p como uma forma válida de argumento, e não é. Esse tipo de raciocínio incorreto é chamado de **falácia da afirmação da conclusão.**

EXEMPLO 10 O seguinte argumento é válido?

> Se fizer todos os exercícios deste livro, então você terá aprendido matemática discreta. Você aprendeu matemática discreta.
>
> Portanto, você fez todos os exercícios deste livro.

Solução: Seja p a proposição "Você fez todos os exercícios deste livro". Seja q a proposição "Você aprendeu matemática discreta". Então este argumento é da forma: se $p \rightarrow q$ e q, então p. Esse é um exemplo de um argumento incorreto que usa a falácia da afirmação da conclusão. Pois é possível você aprender matemática discreta de maneira diferente sem ter de fazer todos os exercícios deste livro. (Você pode aprender matemática discreta lendo, assistindo a palestras, fazendo muitos, mas não todos os exercícios e assim por diante.) ◀

A proposição $[(p \rightarrow q) \wedge \neg p] \rightarrow \neg q$ não é uma tautologia, pois é falsa quando p é falsa e q é verdadeira. Muitos argumentos usam isso incorretamente como se fosse uma regra de inferência. Esse tipo de raciocínio incorreto é chamado de **falácia da negação das hipóteses.**

EXEMPLO 11 Sejam p e q as proposições do Exemplo 10. Se a sentença condicional $p \rightarrow q$ é verdadeira, e $\neg p$ é verdadeira, é correto concluir que $\neg q$ é verdadeira? Em outras palavras, é correto assumir que você não aprende matemática discreta se você não fizer todos os exercícios do livro, assumindo que se você resolver todos os problemas do livro, então você terá aprendido matemática discreta?

Solução: É possível que você aprenda matemática discreta mesmo que você não faça todos os exercícios do livro. Esse argumento incorreto é da forma $p \rightarrow q$ e $\neg p$ implica $\neg q$, que é um exemplo de falácia da negação das hipóteses. ◀

Regras de Inferência para Sentenças Quantificadas

Discutimos regras de inferência para proposições. Vamos agora descrever algumas regras importantes de inferência para sentenças que envolvem quantificadores. Essas regras de inferência são usadas extensivamente em argumentos matemáticos, freqüentemente sem menção explícita.

Instanciação universal é a regra de inferência usada para concluir que $P(c)$ é verdadeira, em que c é um elemento particular do domínio, quando é dada a premissa $\forall x P(x)$. A instanciação universal é usada quando concluímos da sentença "Todas as mulheres são discretas" que "Maria é discreta", em que Maria é um elemento do domínio das mulheres.

Generalização universal é a regra de inferência que diz que $\forall x P(x)$ é verdadeira, dada como premissa que $P(c)$ é verdadeira para todos os elementos c do domínio. A generalização universal é usada quando mostramos que $\forall x P(x)$ é verdadeira tomando um elemento arbitrário c do domínio e mostrando que $P(c)$ é verdadeira. O elemento c que selecionamos deve ser um elemento do domínio arbitrário, e não um específico. Ou seja, quando concluímos de $\forall x P(x)$ a existência de um elemento c no domínio, temos o controle sobre c e não podemos fazer nenhuma outra conclusão sobre c que não seja pertencente ao domínio. A generalização universal é usada implicitamente em muitas demonstrações e é raro ser mencionada explicitamente. No entanto, o erro de fazer conclusões sem garantia sobre um elemento arbitrário c quando a generalização universal é usada é também comum em raciocínios incorretos.

Instanciação existencial é a regra que nos permite concluir que existe um elemento c no domínio para o qual $P(c)$ é verdadeira se sabemos que $\exists x P(x)$ é verdadeira. Não podemos selecionar um valor arbitrário de c aqui, mas deve ser um c para o qual $P(c)$ é verdadeira. Usualmente não temos conhecimento sobre qual c é, apenas que ele existe. Como ele existe, podemos lhe dar um nome (c) e continuar nosso argumento.

Generalização existencial é a regra de inferência que é usada para concluir que $\exists x P(x)$ é verdadeira quando um elemento particular c com $P(c)$ verdadeira é conhecido. Ou seja, se conhecemos um elemento c no domínio para o qual $P(c)$ é verdadeira, então sabemos que $\exists x P(x)$ é verdadeira.

Resumimos essas regras de inferência na Tabela 2. Vamos ilustrar como uma dessas regras de inferência para sentenças quantificadas é usada no Exemplo 12.

EXEMPLO 12 Mostre que as premissas "Todos os alunos da classe de matemática discreta estão tendo um curso de ciência da computação" e "Maria é uma estudante dessa classe" implica a conclusão "Maria está freqüentando um curso de ciência da computação".

TABELA 2 Regras de Inferência para Sentenças Quantificadas.

Regra de Inferência	*Nome*
$\forall x P(x)$ $\therefore P(c)$	Instanciação universal
$P(c)$ para um c arbitrário $\therefore \forall x P(x)$	Generalização universal
$\exists x P(x)$ $\therefore P(c)$ para algum elemento c	Instanciação existencial
$P(c)$ para algum elemento c $\therefore \exists x P(x)$	Generalização existencial

Solução: Seja $D(x)$ a sentença "x está na classe de matemática discreta" e seja $C(x)$ a sentença "x está freqüentando um curso de ciência da computação." Então, as premissas são $\forall x(D(x) \rightarrow C(x))$ e D(Maria). E a conclusão é C(Maria).

Exemplos Extras

Os seguintes passos podem ser utilizados para estabelecer a conclusão a partir das premissas.

Passo	Razão
1. $\forall x(D(x) \rightarrow C(x))$	Premissa
2. D(Maria) $\rightarrow C$(Maria)	Instanciação Universal de (1)
3. D(Maria)	Premissa
4. C(Maria)	*Modus ponens* a partir de (2) e (3)

◀

EXEMPLO 13 Mostre que as premissas "Um estudante desta classe não tem lido o livro" e "Todos nesta classe passaram no primeiro exame" implica a conclusão "Alguém passou no primeiro exame sem ter lido o livro".

Solução: Sejam $C(x)$ a sentença "x está nesta classe", $B(x)$ a sentença "x não tem lido o livro" e $P(x)$ a sentença "x passou no primeiro exame". As premissas são $\exists x(C(x) \wedge \neg B(x))$ e $\forall x(C(x) \rightarrow P(x))$. A conclusão é $\exists x(P(x) \wedge \neg B(x))$. Estes passos podem ser utilizados para estabelecer a conclusão a partir das premissas.

Passo	Razão
1. $\exists x(C(x) \wedge \neg B(x))$	Premissa
2. $C(a) \wedge \neg B(a)$	Instanciação existencial a partir de (1)
3. $C(a)$	Simplificação a partir de (2)
4. $\forall x(C(x) \rightarrow P(x))$	Premissa
5. $C(a) \rightarrow P(a)$	Instanciação universal a partir de (4)
6. $P(a)$	*Modus ponens* a partir (3) e (5)
7. $\neg B(a)$	Simplificação a partir de (2)
8. $P(a) \wedge \neg B(a)$	Conjunção a partir de (6) e (7)
9. $\exists x(P(x) \wedge \neg B(x))$	Generalização existencial a partir de (8)

◀

Combinando Regras de Inferência para Proposições e Sentenças Quantificadas

Desenvolvemos regras de inferência para proposições e para sentenças quantificadas. Note que em nossos argumentos nos exemplos 12 e 13 usamos tanto instanciação universal, uma regra de inferência para sentenças quantificadas, quanto *modus ponens*, uma regra de inferência para a lógica proposicional. Vamos freqüentemente precisar usar essa combinação de regras de inferência. Como instanciação universal e *modus ponens* são usadas freqüentemente juntas, essa combinação de regras é costumeiramente chamada de ***modus ponens* universal**. Essa regra nos diz que se $\forall x(P(x) \rightarrow Q(x))$ é verdadeira, e se $P(a)$ é verdadeira para algum elemento particular a no domínio do quantificador universal, então $Q(a)$ deve ser verdadeira. Para ver isso, note que, por instanciação universal, $P(a) \rightarrow Q(a)$ é verdadeira. Então, por *modus ponens*, $Q(a)$ deve também ser verdadeira. Podemos descrever o *modus ponens* universal como se segue:

$$\begin{array}{l} \forall x(P(x) \rightarrow Q(x)) \\ \underline{P(a), \text{ em que } a \text{ é um elemento particular do domínio}} \\ \therefore Q(a) \end{array}$$

O *modus ponens* universal é comumente usado em argumentos matemáticos. Isso é ilustrado no Exemplo 14.

EXEMPLO 14 Assuma que "Para todo inteiro positivo n, se n é maior que 4, então n^2 é menor que 2^n" é verdadeira. Use o *modus ponens* universal para mostrar que $100^2 < 2^{100}$.

Solução: Seja $P(n)$ a sentença "$n > 4$" e $Q(n)$ a sentença "$n^2 < 2^n$ ". A sentença "Para todo inteiro positivo n, se n é maior que 4, então n^2 é menor que 2^n" pode ser representada por $\forall n(P(n) \rightarrow Q(n))$, em que o domínio consiste em todos os inteiros positivos. Estamos assumindo que $\forall n\,(P(n) \rightarrow Q(n))$ é verdadeira. Note que $P(100)$ é verdadeira, pois $100 > 4$. Então segue por *modus ponens* universal que $Q(n)$ é verdadeira, explicitamente $100^2 < 2^{100}$. ◀

Outra combinação muito utilizada de regra de inferência para lógica com uma regra de inferência para sentenças quantificadas é o ***modus tollens* universal.** *Modus tollens* universal combina a instanciação universal e o *modus tollens* e pode ser expresso por:

$\forall x(P(x) \rightarrow Q(x))$
$\neg Q(a)$, em que a é um elemento particular no domínio
$\therefore \neg P(a)$

Deixamos a verificação do *modus tollens* universal para o leitor (veja o Exercício 25). O Exercício 26 desenvolve combinações adicionais de regras de inferência na lógica proposicional e sentenças quantificadas.

Exercícios

1. Encontre a forma de argumento para o argumento dado e determine se é válido. Podemos inferir que a conclusão é verdadeira se as premissas forem verdadeiras?

Se Sócrates é humano, então Sócrates é mortal.
Sócrates é humano.
∴ Sócrates é mortal.

2. Encontre a forma de argumento para o argumento dado e determine se é válido. Podemos inferir que a conclusão é verdadeira se as premissas forem verdadeiras?

Se George não tem oito patas, então ele não é um inseto.
George é um inseto.
∴ George tem oito patas.

3. Qual a regra de inferência usada em cada um dos argumentos abaixo?

a) Alice é graduada em matemática. Por isso, Alice é graduada ou em matemática ou em ciência da computação.

b) Jerry é um graduado em matemática e em ciência da computação. Por isso, Jerry é um graduado em matemática.

c) Se o dia estiver chuvoso, então a piscina estará fechada. O dia está chuvoso. Por isso, a piscina está fechada.

d) Se nevar hoje, a universidade estará fechada. A universidade não está fechada hoje. Por isso, não nevou hoje.

e) Se eu for nadar, então eu ficarei no sol por muito tempo. Se eu ficar no sol por muito tempo, então eu me queimarei. Por isso, se eu for nadar, eu me queimarei.

4. Qual a regra de inferência utilizada em cada um dos argumentos a seguir?

a) Cangurus vivem na Austrália e são marsupiais. Por isso, cangurus são marsupiais.

b) Ou está mais quente que 100 graus hoje ou a poluição é perigosa. Está menos de 100 graus lá fora hoje. Por isso, a poluição é perigosa.

c) Linda é uma excelente nadadora. Se Linda é uma excelente nadadora, então ela pode trabalhar como salva-vidas. Por isso, Linda pode trabalhar como salva-vidas.

d) Steve trabalhará em uma indústria de computadores neste verão. Por isso, neste verão ele trabalhará em uma indústria de computadores ou ele será um desocupado na praia.

e) Se eu trabalhar a noite toda nesta tarefa de casa, então posso resolver todos os exercícios. Se eu resolver todos os exercícios, eu entenderei o material. Por isso, se eu trabalhar à noite nesta tarefa, então eu entenderei o material.

5. Use as regras de inferência para mostrar que as hipóteses "Randy trabalha muito", "Se Randy trabalha muito, então ele é um garoto estúpido" e "Se Randy é um garoto estúpido, então ele não conseguirá o emprego" implicam a conclusão "Randy não conseguirá o emprego".

6. Use as regras de inferência para mostrar que as hipóteses "Se não chove ou não tem neblina, então a competição de vela acontecerá e a apresentação de salvamento continuará", "Se a competição de vela é mantida, então o troféu será conquistado" e "O troféu não foi conquistado" implicam a conclusão "Choveu".

7. Quais regras de inferência são utilizadas no famoso argumento abaixo?

"Todos os homens são mortais. Sócrates é um homem. Por isso, Sócrates é mortal."

8. Quais as regras de inferência utilizadas no argumento abaixo? "Nenhum homem é uma ilha. Manhattan é uma ilha. Por isso, Manhattan não é um homem."

9. Para cada grupo de premissas abaixo, qual conclusão ou conclusões relevantes podem ser tiradas? Explique as regras de inferência utilizadas para obter cada conclusão das premissas.
- **a)** "Se eu tiro o dia de folga, chove ou neva." "Eu tirei folga na terça-feira ou na quinta-feira." "Fez sol na terça-feira." "Não nevou na quinta-feira."
- **b)** "Se eu como comida apimentada, então eu tenho sonhos estranhos." "Eu tenho sonhos estranhos quando cai um trovão enquanto eu durmo." "Eu não tive sonhos estranhos."
- **c)** "Eu sou ou esperto ou sortudo." "Eu não tenho sorte." "Se eu tivesse sorte, então eu ganharia na loteria."
- **d)** "Todo graduado em ciência da computação tem seu próprio computador." "Ralph não tem seu próprio computador." "Ana tem seu próprio computador."
- **e)** "O que é bom para as corporações é bom para os Estados Unidos." "O que é bom para os Estados Unidos, é bom para você." "O que é bom para as corporações é você comprar muitas coisas."
- **f)** "Todos os roedores roem sua própria comida." "Ratos são roedores." "Coelhos não roem sua comida." "Morcegos não são roedores."

10. Para cada grupo de premissas abaixo, qual conclusão ou conclusões relevantes podem ser tiradas? Explique as regras de inferência utilizadas para obter cada conclusão das premissas.
- **a)** "Se eu jogo hóquei, então eu fico dolorido no dia seguinte." "Eu uso a hidromassagem se eu estou dolorido." "Eu não usei a hidromassagem."
- **b)** "Se eu trabalho, o dia está ensolarado, total ou parcialmente." "Eu trabalhei segunda-feira passada ou trabalhei sexta-feira passada." "O dia não estava ensolarado na terça-feira." "Estava parcialmente ensolarado na sexta-feira."
- **c)** "Todos os insetos têm seis patas." "Libélulas são insetos." "Aranhas não têm seis patas." "Aranhas comem libélulas."
- **d)** "Todo estudante tem uma conta de Internet." "Homer não tem uma conta de Internet." "Maggie tem uma conta de Internet."
- **e)** "Todas as comidas que são saudáveis não têm um sabor bom." "Tofu é uma comida saudável." "Você come apenas o que tem gosto bom." "Você não come tofu." "Cheeseburgers não são comidas saudáveis."
- **f)** "Eu estou dormindo ou tendo alucinações." "Eu não estou dormindo." "Se eu estou tendo alucinações, eu vejo elefantes correndo na estrada."

11. Mostre que o argumento com as premissas $p_1, p_2, \ldots, p_n$ e a conclusão $q \rightarrow r$ é válido se o argumento com as premissas $p_1, p_2, \ldots, p_n, q$ e a conclusão r é válido.

12. Mostre que o argumento com as premissas $(p \wedge t) \rightarrow (r \vee s)$, $q \rightarrow (u \wedge t)$, $u \rightarrow p$, e $\neg s$ e a conclusão $q \rightarrow r$ é válido, usando o Exercício 11 e depois usando as regras de inferência da Tabela 1.

13. Para cada argumento a seguir, aponte quais regras de inferência foram utilizadas em cada passo.
- **a)** "Doug, um estudante desta sala, sabe como escrever programas em Java. Todos que sabem como escrever programas em Java conseguem um emprego bem remunerado. Por isso, alguém nesta sala pode conseguir um emprego bem remunerado."
- **b)** "Alguém nesta sala gosta de ver baleias. Toda pessoa que gosta de ver baleias se preocupa com a poluição no mar. Por isso, há uma pessoa na sala que se preocupa com a poluição marinha."
- **c)** "Cada um dos 93 estudantes nesta classe possuem seu próprio computador. Todos que possuem seu próprio computador podem usar um programa de processamento de palavras. Por isso, Zeke, um estudante da sala, pode usar um programa de processamento."
- **d)** "Todos em Nova Jersey vivem a 50 milhas do oceano. Alguém que mora em Nova Jersey nunca viu o oceano. Por isso, alguém que mora a 50 milhas do oceano nunca o viu."

14. Para cada argumento abaixo, aponte quais regras de inferência foram utilizadas em cada passo.
- **a)** "Linda, uma estudante desta sala, tem um conversível vermelho. Todo mundo que tem um conversível vermelho tem pelo menos uma multa por excesso de velocidade. Por isso, alguém nesta sala tem uma multa por excesso de velocidade."
- **b)** "Cada um dos cinco colegas de quarto, Melissa, Aaron, Ralph, Veneesha e Keeshawn, freqüentou um curso de matemática discreta. Todo estudante que freqüentou um curso de matemática discreta pode freqüentar um curso de algoritmo. Por isso, todos os cinco colegas de quarto podem freqüentar um curso de algoritmo no próximo ano."
- **c)** "Todos os filmes produzidos por John Sayles são maravilhosos. John Sayles produziu um filme sobre mineiros de carvão mineral. Por isso, há um filme maravilhoso sobre mineiros de carvão."
- **d)** "Há alguém nesta sala que foi à França. Todos que vão à França visitam o Louvre. Por isso, alguém nesta sala visitou o Louvre."

15. Determine se cada um dos argumentos abaixo é correto ou incorreto e explique o porquê.
- **a)** "Todos os estudantes nesta sala entendem lógica. Xavier é um estudante desta sala. Por isso, Xavier entende lógica."
- **b)** "Todo graduando em ciência da computação faz matemática discreta. Natasha está fazendo matemática discreta. Por isso, Natasha é uma graduanda em ciência da computação."
- **c)** "Todos os papagaios gostam de frutas. Meu passarinho de estimação não é um papagaio. Por isso, meu passarinho de estimação não gosta de frutas."
- **d)** "Todos que comem granola todo dia são saudáveis. Linda não é saudável. Por isso, Linda não come granola todos os dias."

16. Determine se cada um dos argumentos abaixo é correto ou incorreto e explique o porquê.
- **a)** "Todos que são matriculados na universidade moram em um dormitório. Mia nunca morou em um dormitório. Por isso, Mia não está matriculada na universidade."
- **b)** "Um carro conversível é bom de dirigir. O carro de Isaac não é um conversível. Por isso, o carro de Isaac não é bom de dirigir."

c) "Quincy gosta de todos os filmes de ação. Quincy gosta do filme *Eight Men Out*. Por isso, *Eight Men Out* é um filme de ação."

d) "Todos os homens que pescam lagostas armam pelo menos uma dúzia de armadilhas. Hamilton é um pescador de lagostas. Por isso, Hamilton arma pelo menos uma dúzia de armadilhas."

17. O que está errado neste argumento? Considere $H(x)$ como "x é feliz". Dada a premissa $\exists x\ H(x)$, concluímos que H(Lola). Por isso, Lola é feliz.

18. O que está errado neste argumento? Considere $S(x, y)$ como "x é mais baixo que y". Dada a premissa $\exists s\ S(s,$ Max), segue que S(Max, Max). Então, pela generalização existencial, temos que $\exists x\ S(x, x)$, ou seja, alguém é mais baixo que si próprio.

19. Determine se cada um dos argumentos abaixo é válido. Se um argumento estiver correto, qual regra de inferência foi utilizada? Se não, quais erros lógicos foram cometidos?

a) Se n é um número real, tal que $n > 1$, então $n^2 > 1$. Suponha que $n^2 > 1$. Então $n > 1$.

b) Se n é um número real com $n > 3$, então $n^2 > 9$. Suponha que $n^2 \leq 9$. Então $n \leq 3$.

c) Se n é um número real com $n > 2$, então $n^2 > 4$. Suponha que $n \leq 2$. Então $n^2 \leq 4$.

20. Determine se os argumentos abaixo são válidos.

a) Se x é um número real positivo, então x^2 é um número real positivo. Por isso, se a^2 é positivo, em que a é um número real, então a é um número real positivo.

b) Se $x^2 \neq 0$, em que x é um número real, então $x \neq 0$. Considere a como um número real com $a^2 \neq 0$; então $a \neq 0$.

21. Quais regras de inferência foram utilizadas para estabelecer a conclusão do argumento de Lewis Carroll descrito no Exemplo 26 da Seção 1.3?

22. Quais regras de inferência foram utilizadas para estabelecer a conclusão do argumento de Lewis Carroll descrito no Exemplo 27 da Seção 1.3?

23. Identifique o(s) erro(s) neste argumento que supostamente mostra(m) que se $\exists x P(x) \wedge \exists x Q(x)$ é verdadeira, então $\exists x(P(x) \wedge Q(x))$ é verdadeira.

1. $\exists x\ P(x) \wedge \exists x\ Q(x)$	Premissa
2. $\exists x\ P(x)$	Simplificação de (1)
3. $P(c)$	Instanciação Existencial de (2)
4. $\exists x\ Q(x)$	Simplificação de (1)
5. $Q(c)$	Instanciação Existencial de (4)
6. $P(c) \wedge Q(c)$	Conjunção de (3) e (5)
7. $\exists x(P(x) \wedge Q(x))$	Generalização Existencial

24. Identifique o(s) erro(s) neste argumento que supostamente mostra(m) que se $\forall x(P(x) \vee Q(x))$ é verdadeira, então $\forall x\ P(x) \vee \forall x\ Q(x)$ é verdadeira.

1. $\forall x\ P(x) \vee Q(x))$	Premissa
2. $P(c) \vee Q(c)$	Instanciação universal de (1)
3. $P(c)$	Simplificação de (2)
4. $\forall x P(x)$	Generalização universal de (3)
5. $Q(c)$	Simplificação de (2)
6. $\forall x Q(x)$	Generalização universal de (5)
7. $\forall x(P(x) \vee \forall x Q(x))$	Conjunção de (4) e (6)

25. Justifique a regra universal de *modus tollens* mostrando que as premissas $\forall x(P(x) \rightarrow Q(x))$ e $\neg Q(a)$ para um elemento em particular a no domínio, implica $\neg P(a)$.

26. Justifique a regra da **transitividade universal**, que declara que se $\forall x(P(x) \rightarrow Q(x))$ e $\forall x\,(Q(x) \rightarrow R(x))$ são verdadeiras, então $\forall x\ (P(x) \rightarrow R(x))$ é verdadeira, em que os domínios de todos os quantificadores são os mesmos.

27. Use as regras de inferência para mostrar que se $\forall x(P(x) \rightarrow (Q(x) \wedge S(x)))$ e $\forall x(P(x) \wedge R(x))$ são verdadeiras, então $\forall x(R(x) \wedge S(x))$ é verdadeira.

28. Use as regras de inferência para mostrar que se $\forall x(P(x) \vee Q(x))$ e $\forall x(\neg P(x) \wedge Q(x)) \rightarrow R(x))$ são verdadeiras, então $\forall x(\neg R(x) \rightarrow P(x))$ é também verdadeira, em que os domínios de todos os quantificadores são os mesmos.

29. Use as regras de inferência para mostrar que se $\forall x(P(x) \vee Q(x))$, $\forall x(\neg Q(x) \vee S(x))$, $\forall x(R(x) \rightarrow \neg S(x))$, e $\exists x \neg P(x)$ são verdadeiras, então $\exists x \neg R(x)$ é verdadeira.

30. Use a resolução para mostrar que as hipóteses "Allen é um garoto mau ou Hillary é uma boa garota" e "Allen é um bom garoto ou David é feliz" implica a conclusão "Hillary é uma boa garota ou David é feliz".

31. Use a resolução para mostrar que as hipóteses "Não está chovendo ou Ivete tem sua sombrinha", "Ivete não tem uma sombrinha ou ela não quer se molhar" e "Está chovendo ou Ivete não se molha" implica que "Ivete não se molha".

32. Mostre que a equivalência $p \wedge \neg\ p \equiv \mathbf{F}$ pode ser derivada usando a resolução junto com o fato de que uma proposição condicional com uma hipótese falsa é verdadeira. [*Dica:* Considere $q = r = \mathbf{F}$ na resolução.]

33. Use a resolução para mostrar que uma proposição composta $(p \vee q) \wedge (\neg p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee \neg q)$ não é satisfatória.

***34.** O Problema de Lógica, tirado da *Prova FBF, O Jogo da Lógica*, tem essas duas suposições:

1. "Lógica é difícil ou não muitos estudantes gostam de lógica."
2. "Se matemática é fácil, então lógica não é difícil."

Transcrevendo essas suposições em proposições que envolvam variáveis proposicionais e conectivos lógicos, determine se cada uma das seguintes conclusões é válida para as suposições:

a) Matemática não é fácil, se muitos estudantes gostam de lógica.

b) Poucos estudantes gostam de lógica, se matemática não é fácil.

c) Matemática não é fácil ou lógica é difícil.

d) Lógica não é difícil ou matemática não é fácil.

e) Se poucos estudantes gostam de lógica, então matemática não é fácil ou lógica não é difícil.

***35.** Determine se este argumento, tirado do Kalish e Montague [KaMo64], é válido.

> Se o Super-homem era capaz e tinha vontade de combater o mal, ele seria benevolente. Se o Super-homem não fosse capaz de combater o mal, ele seria impotente; se ele não tivesse vontade de combater o mal, ele seria malevolente. O Super-homem não combate o mal. Se o Super-homem existe, ele é ou impotente ou malevolente. Por isso, o Super-homem não existe.

1.6 Introdução a Demonstrações

Introdução

Nesta seção, introduziremos a noção de demonstração e descreveremos métodos para a construção de demonstrações. Uma demonstração é um argumento válido que estabelece a verdade de uma sentença matemática. Uma demonstração pode usar as hipóteses do teorema, se existirem, axiomas assumidos com verdade e teoremas demonstrados anteriormente. Usando esses ingredientes e regras de inferência, o passo final da demonstração estabelece a verdade da sentença que está sendo demonstrada.

Em nossa discussão, vamos nos mover de demonstrações formais de teoremas até demonstrações mais informais. Os argumentos que introduzimos na Seção 1.5 para demonstrar que sentenças que envolvem proposições e sentenças quantificadas são verdadeiras sob demonstrações formais, se todos os passos são dados, e as regras para cada passo do argumento são também dadas. No entanto, demonstrações formais de teoremas muito comuns podem ser extremamente longas e difíceis de fazer. Na prática, as demonstrações dos teoremas feitas por humanos são na sua maioria **demonstrações informais**, em que mais de uma regra de inferência pode ser utilizada em cada passo, passos podem ser pulados, axiomas são assumidos e as regras de inferência utilizadas em cada passo não são explicitamente demonstradas. Demonstrações informais podem explicar aos humanos por que teoremas são verdadeiros, enquanto computadores só se contentam quando produzem uma demonstração formal usando sistemas de raciocínio automático.

Os métodos de demonstrações discutidos neste capítulo são importantes não só porque são utilizados para demonstrar teoremas, mas também pelas muitas aplicações em ciência da computação. Essas aplicações incluem verificar se programas de computador são corretos, estabelecendo se sistemas de operação são seguros, fazendo inferência em inteligência artificial, mostrando que sistemas de especificações são consistentes, e assim por diante. Conseqüentemente, compreender as técnicas utilizadas em demonstrações é essencial tanto para matemática quanto para ciência da computação.

Alguma Terminologia

Formalmente, um **teorema** é uma sentença que se pode demonstrar que é verdadeira. Em escrita matemática, o termo teorema é usualmente reservado para as sentenças que são consideradas com alguma importância. Teoremas menos importantes são comumente chamados de **proposições**. (Teoremas podem ser também referidos como **fatos** ou **resultados**.) Um teorema pode ser uma quantificação universal de uma sentença condicional com uma ou mais premissas e uma conclusão. No entanto, pode ser outro tipo de sentença lógica, como os exemplos vão mostrar, mais tarde, neste capítulo. Nós demonstramos que um teorema é verdadeiro com uma **demonstração**. Uma demonstração é um argumento válido que estabelece a verdade de um teorema. As sentenças utilizadas na demonstração podem incluir **axiomas** (ou **postulados**), os quais são sentenças que assumimos ser verdadeiras (por exemplo, veja Apêndice 1 com axiomas para os números reais), as premissas do teorema, se existirem, e teoremas previamente provados. Axiomas podem ser descritos usando termos primitivos que não requerem definição, mas todos os outros termos utilizados em teoremas e suas demonstrações devem ser definidos. Regras de inferência, juntamente com as definições dos termos, são utilizadas para chegar a conclusões a partir de outras afirmações, unindo os passos da demonstração. Na prática, o passo final de uma demonstração é usualmente a conclusão do teorema. No entanto, para esclarecer, vamos freqüentemente retomar a sentença do teorema como o passo final de uma demonstração.

Um teorema menos importante que nos ajuda em uma demonstração de outros resultados é chamado de **lema** (plural *lemas* ou *lemata*). Demonstrações complicadas são usualmente mais fáceis de entender quando elas são demonstradas utilizando-se uma série de lemas, em que cada lema é demonstrado individualmente. Um **corolário** é um teorema que pode ser estabelecido diretamente de um teorema que já foi demonstrado. Uma **conjectura** é uma sentença que inicialmente é proposta como verdadeira, usualmente com base em alguma evidência parcial, um argumento heurístico ou a intuição de um perito. Quando uma demonstração de uma conjectura é achada, a conjectura se torna um teorema. Muitas vezes são verificadas que conjecturas são falsas, portanto elas não são teoremas.

Entendendo como Teoremas São Descritos

Exemplos Extras

Antes de introduzir métodos para demonstrar teoremas, precisamos entender como teoremas matemáticos são expostos. Muitos teoremas dizem que essa propriedade é assegurada para todos os elementos em um domínio, como os inteiros ou os números reais. Embora as sentenças precisas desses teoremas necessitem da inclusão de um quantificador universal, a convenção em matemática é omiti-la. Por exemplo, a sentença

"Se $x > y$, em que x e y são números reais positivos, então $x^2 > y^2$."

significa que

"Para todos os números reais positivos x e y, se $x > y$, então $x^2 > y^2$."

Entretanto, quando teoremas desse tipo são demonstrados, a propriedade da instanciação universal é freqüentemente usada sem ser explicitamente mencionada. O primeiro passo da demonstração usualmente envolve selecionar um elemento geral do domínio. Os passos subseqüentes mostram que esse elemento tem a propriedade em questão. Finalmente, a generalização universal implica que o teorema é válido para todos os membros do domínio.

Métodos de Demonstrações de Teoremas

Auto-avaliação

Vamos agora mudar nossa atenção para demonstração de teoremas matemáticos. Demonstrar teoremas pode ser difícil. Vamos precisar de toda a munição que tivermos para nos ajudar a demonstrar resultados diferentes. Vamos, então, introduzir uma bateria de métodos de demonstrações diferentes. Esses métodos podem se tornar parte de nosso repertório para demonstrar teoremas.

Para demonstrar um teorema da forma $\forall x(P(x) \rightarrow Q(x))$, nosso objetivo é mostrar que $P(c) \rightarrow Q(c)$ é verdadeira, em que c é um elemento arbitrário do domínio, e então aplicar a generalização universal. Nesta demonstração, precisamos mostrar que uma sentença condicional é verdadeira. Por isso, focalizaremos métodos que demonstram que condicionais são verdadeiras. Lembre-se de que $p \rightarrow q$ é verdadeira, a menos que p seja verdadeira e q seja falsa. Note que para a sentença $p \rightarrow q$ ser demonstrada, é necessário apenas mostrar que q é verdadeira se p é verdadeira. A seguinte discussão nos dará as técnicas mais comuns para demonstrar sentenças condicionais. Mais tarde vamos discutir métodos para demonstrar outros tipos de sentenças. Nesta seção e na Seção 1.7, vamos desenvolver um arsenal de muitas técnicas diferentes de demonstração, que podem ser usadas para demonstrar uma grande variedade de teoremas.

Quando você ler demonstrações, encontrará freqüentemente as palavras "obviamente" ou "claramente". Essas palavras indicam que passos foram omitidos e que o autor espera que o leitor seja capaz de fazê-los. Infelizmente, assumir isso nem sempre é interessante, pois os leitores não são todos capazes de fazer os passos nesses buracos das demonstrações. Vamos assiduamente tentar não usar essas palavras e não omitir muitos passos. No entanto, se concluirmos todos os passos em demonstrações, nossas demonstrações serão com freqüência exaustivamente longas.

Demonstrações Diretas

Uma **demonstração direta** de uma sentença condicional $p \rightarrow q$ é construída quando o primeiro passo é assumir que p é verdadeira; os passos subseqüentes são construídos utilizando-se regras de inferência, com o passo final mostrando que q deve ser também verdadeira. Uma demonstração direta mostra que uma sentença condicional $p \rightarrow q$ é verdadeira mostrando que p é verdadeira, então q deve ser verdadeira, de modo que a combinação p verdadeira e q falsa nunca ocorre. Em uma demonstração direta, assumimos que p é verdadeira e usamos axiomas, definições e teoremas previamente comprovados, junto com as regras de inferência, para mostrar que q deve ser também verdadeira. Você verá que demonstrações diretas de muitos resultados são construídas de

maneira direta, com uma seqüência óbvia de passos que levam da hipótese à conclusão. No entanto, demonstrações diretas algumas vezes requerem *insights* particulares e podem ser bastante astuciosas. As primeiras demonstrações diretas que vamos apresentar aqui são bastante óbvias; mais tarde veremos algumas menos óbvias.

Vamos dar muitos exemplos de demonstrações diretas. Mas, antes de darmos o primeiro exemplo, precisamos de uma definição.

DEFINIÇÃO 1 O inteiro n é *par* se existe um inteiro k tal que $n = 2k$, e n é *ímpar* se existe um inteiro k tal que $n = 2k + 1$. (Note que um inteiro é sempre par ou ímpar e nenhum inteiro é par e ímpar.)

EXEMPLO 1 Dê uma demonstração direta do teorema "Se n é um número inteiro ímpar, então n^2 *é ímpar*".

Solução: Note que este teorema diz $\forall n P((n) \rightarrow Q(n))$, em que $P(n)$ é "n é um inteiro ímpar" e $Q(n)$ é "n^2 *é ímpar*". Como dissemos, vamos seguir a convenção matemática usual para demonstrações, mostrando que $P(n)$ implica $Q(n)$, e não usando explicitamente instanciação universal. Para começar uma demonstração direta desse teorema, vamos assumir que a hipótese dessa sentença condicional é verdadeira, ou seja, assumimos que n é ímpar. Pela definição de número ímpar, temos que $n = 2k + 1$, em que k é algum inteiro. Queremos demonstrar que n^2 é também ímpar. Podemos elevar ao quadrado ambos os membros da equação $n = 2k + 1$ para obter uma nova equação que expresse n^2. Quando fizermos isso, teremos $n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$. Pela definição de inteiro ímpar, concluímos que n^2 é ímpar (ele é um a mais que o dobro de um inteiro). Conseqüentemente, provamos que se n é um número inteiro ímpar, então n^2 é ímpar. ◀

Exemplos Extras

EXEMPLO 2 Dê uma demonstração direta de que se m e n são ambos quadrados perfeitos, então nm também é um quadrado perfeito. (Um inteiro a é um **quadrado perfeito** se existe um inteiro b tal que $a = b^2$.)

Solução: Para produzir uma demonstração direta desse teorema, assumimos que a hipótese dessa condicional é verdadeira, ou seja, assumimos que m e n são ambos quadrados perfeitos. Pela definição de quadrado perfeito, segue-se que existem inteiros s e t tal que $m = s^2$ e $n = t^2$. O objetivo da demonstração é mostrar que mn também deve ser um quadrado perfeito quando m e n o são; olhando adiante, vemos como podemos mostrar isto apenas multiplicando as duas equações $m = s^2$ e $n = t^2$. Isso mostra que $mn = s^2t^2$, o que implica que $mn = (st)^2$ (usando comutatividade e associatividade da multiplicação). Pela definição de quadrado perfeito, segue que mn também é um quadrado perfeito, pois é o quadrado de st, o qual também é um inteiro. Demonstramos que se m e n são ambos quadrados perfeitos, então mn também é um quadrado perfeito.

Demonstração por Contraposição

Demonstrações diretas vão da hipótese do teorema à sua conclusão. Elas começam com as premissas, continuam com uma seqüência de deduções e com a conclusão. No entanto, vamos ver que tentar fazer demonstrações diretas freqüentemente não tem um bom final. Precisamos de outros métodos para demonstrar teoremas da forma $\forall x(P(x) \rightarrow Q(x))$. Demonstrações de teoremas desse tipo que não são demonstrações diretas, ou seja, que não seguem das hipóteses e terminam com a conclusão, são chamadas de **demonstrações indiretas**.

Uma demonstração indireta extremamente usada é conhecida como **demonstração por contraposição**. Demonstrações por contraposição fazem uso do fato de que a sentença condicional $p \rightarrow q$ é equivalente a sua contrapositiva, $\neg q \rightarrow \neg p$, isto significa que uma sentença condicional

$p \rightarrow q$ pode ser demonstrada mostrando que sua contrapositiva, $\neg q \rightarrow \neg p$, é verdadeira. Em uma demonstração por contraposição de $p \rightarrow q$, vamos tomar $\neg q$ como uma hipótese, e usando axiomas, definições e teoremas previamente demonstrados, juntamente com regras de inferência, mostramos que $\neg p$ deve ser verdadeira.Vamos ilustrar demonstração por contraposição com dois exemplos. Esses exemplos mostram que demonstrações por contraposição podem ser bem-sucedidas quando não é fácil achar demonstrações diretas.

EXEMPLO 3 Demonstre que se n é um inteiro e $3n + 2$ é ímpar, então n é ímpar.

Exemplos Extras

*Solução:*Primeiro vamos olhar para uma demonstração direta. Para construir uma demonstração direta, devemos assumir que $3n + 2$ é um número ímpar. Isso significa que $3n + 2 = 2k + 1$ para algum inteiro k. Podemos usar esse fato para mostrar que n é ímpar? Vemos que $3n + 1 = 2k$, mas não parece ter algum meio direto para concluir que n é ímpar. Como nossa tentativa com demonstração direta falhou, vamos tentar uma demonstração por contraposição.

O primeiro passo em uma demonstração por contraposição é assumir que a conclusão da sentença condicional "Se $3n + 2$ é ímpar, então n é ímpar" é falsa; assumimos que n é par. Então, pela definição de número par, $n = 2k$ para algum inteiro k. Substituindo $2k$ em n, chegamos a $3n + 2 = 3(2k) + 2 = 6k + 2 = 2(3k + 1)$. Isso nos diz que $3n + 2$ é par (pois é um múltiplo de 2), e logo não é ímpar. Isso é a negação da hipótese do teorema. Como a negação da conclusão da sentença condicional implica que a hipótese é falsa, a sentença original é verdadeira. Nossa demonstração por contraposição foi bem-sucedida; demonstramos que se n é um inteiro e $3n + 2$ é ímpar, então n é ímpar. ◀

EXEMPLO 4 Demonstre que se $n = ab$, em que a e b são inteiros positivos, então $a \leq \sqrt{n}$ ou $b \leq \sqrt{n}$.

Solução: Como não há um meio óbvio de mostrar que $a \leq \sqrt{n}$ ou $b \leq \sqrt{n}$ diretamente da equação $n = ab$, em que a e b são inteiros positivos, vamos tentar uma demonstração por contraposição.

O primeiro passo nesta demonstração é assumir que a conclusão da condicional "Se $n = ab$, em que a e b são inteiros positivos, então $a \leq \sqrt{n}$ ou $b \leq \sqrt{n}$ " é falsa. Ou seja, assumir que a sentença $(a \leq \sqrt{n}) \vee (b \leq \sqrt{n})$ é falsa. Usando o significado da disjunção junto com a lei de De Morgan, vemos que isso implica que ambos $a \leq \sqrt{n}$ e $b \leq \sqrt{n}$ são falsas. Isso implica que $a > \sqrt{n}$ e $b > \sqrt{n}$. Podemos multiplicar essas inequações juntas (usando o fato de que se $0 < t$ e $0 < v$, então $su < tv$) para obter $ab > \sqrt{n} \cdot \sqrt{n} = n$. Isso mostra que $ab \neq n$, o que contradiz a sentença $n = ab$.

Como a negação da conclusão da condicional implica que a hipótese é falsa, a sentença condicional original é verdadeira. Nossa demonstração por contraposição foi bem-sucedida; demonstramos que se $n = ab$, em que a e b são inteiros positivos, então $a \leq \sqrt{n}$ ou $b \leq \sqrt{n}$. ◀

DEMONSTRAÇÃO POR VACUIDADE E DEMONSTRAÇÃO POR TRIVIALIZAÇÃO Podemos rapidamente demonstrar que uma sentença condicional $p \rightarrow q$ é verdadeira quando sabemos que p é falsa, pois $p \rightarrow q$ deve ser verdadeira quando p é falsa. Conseqüentemente, se pudermos mostrar que p é falsa, então teremos uma demonstração, chamada de **demonstração por vacuidade**, da condicional $p \rightarrow q$. Demonstrações por vacuidade são empregadas para estabelecer casos especiais de teoremas que dizem que uma condicional é verdadeira para todos os números inteiros positivos [isto é, um teorema do tipo $\forall n P(n)$, em que $P(n)$ é uma função proposicional]. Técnicas de demonstração para esse tipo de teoremas serão discutidas na Seção 4.1.

EXEMPLO 5 Mostre que a proposição $P(0)$ é verdadeira, em que $P(n)$ é "Se $n > 1$, então $n^2 > n$" e o domínio consiste todos os inteiros.

Solução: Note que $P(0)$ é "Se $0 > 1$, então $0^2 > 0$". Podemos mostrar $P(0)$ utilizando-se uma demonstração por vacuidade, pois a hipótese $0 > 1$ é falsa. Isso nos diz que $P(0)$ é automaticamente verdadeira. ◀

Lembre-se: O fato de a conclusão desta sentença condicional, $0^2 > 0$, ser falsa é irrelevante para o valor-verdade da sentença condicional, pois uma condicional com uma hipótese falsa é diretamente verdadeira.

Podemos também demonstrar rapidamente que a condicional $p \rightarrow q$ é verdadeira se sabemos que a conclusão q é verdadeira. Mostrar que q é verdadeira faz com que $p \rightarrow q$ deva ser também verdadeira. Uma demonstração de $p \rightarrow q$ que usa o fato de que q é verdadeira é chamada de **demonstração por trivialização**. Demonstrações por trivialização são freqüentemente usadas e de grande importância quando demonstramos casos especiais de teoremas (veja a discussão de demonstrações por casos na Seção 1.7) e em indução matemática, que é uma técnica de demonstração discutida na Seção 4.1.

EXEMPLO 6 Seja $P(n)$ a proposição "Se a e b são inteiros positivos com $a \geq b$, então $a^n \geq b^n$", em que o domínio consiste em todos os inteiros. Mostre que $P(0)$ é verdadeira.

Solução: A proposição $P(0)$ é "Se $a \geq b$, então $a^0 \geq b^0$". Como $a^0 = b^0 = 1$, a conclusão da condicional é "Se $a \geq b$, então $a^0 \geq b^0$" é verdadeira. Portanto, a sentença condicional, que é $P(0)$, é verdadeira. Este é um exemplo de demonstração por trivialização. Note que a hipótese, a sentença "$a \geq b$", não foi necessária nesta demonstração. ◀

UM POUCO DE ESTRATÉGIA DE DEMONSTRAÇÃO Descrevemos dois importantes métodos para provar teoremas da forma $\forall x(P(x) \rightarrow Q(x))$: demonstração direta e demonstração por contraposição. Também demos exemplos que mostram como cada uma é usada. No entanto, quando você recebe um teorema da forma $\forall x(P(x) \rightarrow Q(x))$, que método você deve tentar usar para demonstrar? Vamos prover algumas regras rápidas aqui; na Seção 1.7 vamos discutir estratégias de demonstração com mais detalhes. Quando queremos demonstrar uma sentença da forma $\forall x(P(x) \rightarrow Q(x))$, primeiro devemos avaliar o que parece ser uma demonstração direta para esta sentença. Comece expandindo as definições da hipótese. Vá raciocinando sobre essas hipóteses, juntamente com os axiomas e os teoremas demonstrados. Se uma demonstração direta não aparecer em nenhuma situação, tente a mesma coisa com a contraposição. Lembre-se de que em uma demonstração por contraposição você assume que a conclusão é falsa e usa uma demonstração direta para mostrar que a hipótese deve ser falsa. Vamos ilustrar essa estratégia nos exemplos 7 e 8. Antes de apresentar nossos próximos exemplos, precisamos de uma definição.

DEFINIÇÃO 2 O número real r é *racional* se existem inteiros p e q com $q \neq 0$, tal que $r = p/q$. Um número real que não é racional é chamado de *irracional*.

EXEMPLO 7 Demonstre que a soma de dois números racionais é um racional. (Note que se incluirmos o quantificador implícito aqui, o teorema que queremos demonstrar é "Para todo número real r e todo real s, se r e s são números racionais, então $r + s$ é racional".)

Exemplos Extras

Solução: Primeiro tentemos uma demonstração direta. Para começar, suponha que r e s são racionais. Da definição de números racionais, segue que existem inteiros p e q, com $q \neq 0$, tal que $r = p/q$, e inteiros t e u, com $u \neq 0$, tal que $s = t/u$. Podemos usar essas informações para mostrar que $r + s$ é racional? O próximo passo é adicionar $r = p/q$ e $s = t/u$, para obter

$$r + s = \frac{p}{q} + \frac{t}{u} = \frac{pu + qt}{qu}.$$

Como $q \neq 0$ e $u \neq 0$, temos que $qu \neq 0$. Conseqüentemente, expressamos $r + s$ como a razão de dois inteiros, $pu + qt$ e qu, em que $qu \neq 0$. Isso significa que $r + s$ é racional. Demonstramos que a soma de dois números racionais é racional; nossa tentativa de achar uma demonstração direta foi bem-sucedida. ◀

EXEMPLO 8 Demonstre que se n é um número inteiro e n^2 *é ímpar*, então n é ímpar.

Solução: Primeiro tentaremos uma demonstração direta. Suponha que n é um inteiro e n^2 é ímpar. Então existe um inteiro k, tal que $n^2 = 2k + 1$. Podemos usar essa informação para mostrar que n é ímpar? Parece que não existe uma saída óbvia para mostrar que n é ímpar, pois, resolvendo a equação em n, temos $n = \pm\sqrt{2k+1}$, que não é interessante de trabalhar.

Como a tentativa de uma demonstração direta não rendeu frutos, tentaremos uma demonstração por contraposição. Tomaremos como hipótese a sentença n não é ímpar. Como todo inteiro é par ou ímpar, isso significa que n é par. E isso implica que existe um inteiro k, tal que $n = 2k$. Para demonstrar o teorema, devemos mostrar que essa hipótese implica a conclusão, ou seja, n^2 não é ímpar, ou ainda, que n^2 é par. Podemos usar a equação $n = 2k$ para determinar isso? Elevando ao quadrado ambos os membros da equação, obtemos $n^2 = 4k^2 = 2(2k^2)$, o que implica que n^2 é também par, pois $n^2 = 2t$, em que $t = 2k^2$. Demonstramos que se n é um número inteiro e n^2 é ímpar, então n é ímpar. Nossa tentativa de encontrar uma demonstração por contraposição foi bem-sucedida. ◀

Demonstração por Contradição

Suponha que queremos demonstrar que uma sentença p é verdadeira. Além disso, suponha que podemos achar uma contradição q tal que $\neg p \rightarrow q$ é verdadeira. Como q é falsa, mas $\neg p \rightarrow q$ é verdadeira, podemos concluir que $\neg p$ é falsa, o que significa que p é verdadeira. Como podemos encontrar uma contradição q que possa nos ajudar a demonstrar que p é verdadeira usando esse raciocínio?

Como a sentença $r \wedge \neg r$ é uma contradição qualquer que seja a proposição r, podemos demonstrar que p é verdadeira se pudermos mostrar que $\neg p \rightarrow (r \wedge \neg r)$ é verdadeira para alguma proposição r. Demonstrações desse tipo são chamadas de **demonstrações por contradição**. Como uma demonstração por contradição não mostra o resultado diretamente, esse é um outro tipo de demonstração indireta. Daremos três exemplos de demonstração por contradição. O primeiro é um exemplo de uma aplicação do princípio da casa dos pombos, uma técnica combinatória que vamos desenvolver profundamente na Seção 5.2.

EXEMPLO 9 Demonstre que ao menos 4 de 22 dias escolhidos devem cair no mesmo dia da semana.

Solução: Seja p a proposição "Ao menos 4 dos 22 dias escolhidos caem no mesmo dia da semana". Suponha que $\neg p$ é verdadeira. Isso significa que no máximo 3 dos 22 dias caem no mesmo dia da semana. Como existem 7 dias na semana, isso implica que no máximo 21 dias podem ser escolhidos, pois, para cada dia da semana, podem ser escolhidos no máximo 3 dias que coincidem no mesmo dia da semana. Isso contradiz a hipótese que afirmava ter 22 dias considerados. Assim, se r é a sentença que diz que 22 dias foram escolhidos, então temos mostrado que $\neg p \rightarrow (r \wedge \neg r)$. Conseqüentemente, sabemos que p é verdadeira. Demonstramos que ao menos 4 de 22 dias escolhidos devem cair no mesmo dia da semana. ◀

Exemplos Extras

EXEMPLO 10 Demonstre que $\sqrt{2}$ é irracional por meio de uma demonstração por contradição.

Solução: Seja p a proposição "$\sqrt{2}$ é irracional". Para começar uma demonstração por contradição, supomos que $\neg p$ é verdadeira. Note que $\neg p$ é a sentença "Não é o caso que $\sqrt{2}$ é irracional", o que diz que $\sqrt{2}$ é racional. Vamos demonstrar que, assumindo que $\neg p$ é verdadeira, chegaremos a uma contradição.

Se $\sqrt{2}$ é racional, existem inteiros a e b tal que $\sqrt{2} = a/b$, em que a e b não têm fator comum (então a fração a/b é irredutível). (Aqui, estamos usando o fato de que todo número racional pode ser escrito em uma fração irredutível.) Como $\sqrt{2} = a/b$, quando ambos os membros da equação são elevados ao quadrado, segue-se que

$$2 = a^2/b^2.$$

Portanto,

$$2b^2 = a^2.$$

Pela definição de número par segue-se que a^2 é par. Podemos usar o fato de que se a^2 é par, então a é par, o qual segue do Exercício 16. Mas se a é par, pela definição de número par, $a = 2c$ para algum inteiro c. Então,

$$2b^2 = 4c^2.$$

Dividindo ambos os membros dessa equação por 2, temos

$$b^2 = 2c^2.$$

Pela definição de par, isso significa que b^2 é par. Novamente usando o fato de que se o quadrado de um inteiro é par, então o inteiro também deve ser par, concluímos que b deve ser par também.

Agora mostramos que ter assumido $\neg p$ nos levou à equação $\sqrt{2} = a/b$, em que a e b não têm fator comum; mas a e b são pares, ou seja, 2 divide ambos os número a e b. Note que a sentença $\sqrt{2} = a/b$, em que a e b não têm fator comum, significa em particular que 2 não divide ambos a e b. Como ter assumido $\neg p$ nos levou à contradição de que 2 divide ambos a e b e 2 não divide ambos a e b, $\neg p$ deve ser falsa. Ou seja, a sentença p, "$\sqrt{2}$ é irracional", é verdadeira. Provamos que $\sqrt{2}$ é irracional. ◀

Demonstração por contradição pode ser usada para demonstrar condicionais. Nessas demonstrações, primeiro assumimos a negação da conclusão. Então, usamos as premissas do teorema e a negação da conclusão para chegar à contradição. (A razão pela qual essas demonstrações são válidas está na equivalência lógica $p \rightarrow q$ e $(p \wedge \neg q) \rightarrow \mathbf{F}$. Para ver que essas sentenças são equivalentes, simplesmente porque cada uma é falsa em exatamente um caso, quando p é verdadeira e q é falsa.)

Note que podemos reescrever uma demonstração por contraposição de uma sentença condicional como uma demonstração por contradição. Em uma demonstração de $p \rightarrow q$ por contraposição, assumimos que $\neg q$ é verdadeira. E, então, mostramos que $\neg p$ também deve ser verdadeira. Para reescrever uma demonstração por contraposição de $p \rightarrow q$ como uma demonstração por contradição, supomos que ambas p e $\neg q$ são verdadeiras. Então, usamos os passos de uma demonstração de $\neg q \rightarrow \neg p$ para mostrar que $\neg p$ é verdadeira. Isso nos leva à contradição $p \wedge \neg p$, completando a demonstração. O Exemplo 11 ilustra como uma demonstração por contraposição de uma condicional pode ser reescrita como uma demonstração por contradição.

EXEMPLO 11 Dê uma demonstração por contradição do teorema "Se $3n + 2$ é ímpar, então n é ímpar".

Solução: Seja p a proposição "$3n + 2$ é ímpar" e q a proposição "n é ímpar". Para construir uma demonstração por contradição, assumimos que ambas p e $\neg q$ são verdadeiras. Ou seja, assumimos que $3n + 2$ é ímpar e que n não o é. Como n não é ímpar, sabemos que é par. Seguindo os passos da solução do Exemplo 3 (uma demonstração por contraposição), podemos mostrar que se n é par, então $3n + 2$ é par. Primeiro, como n é par, existe um inteiro k, tal que $n = 2k$. Isso implica que $3n + 2 = 3(2k) + 2 = 6k + 2 = 2(3k + 1)$. Como $3n + 2$ é $2t$, em que $t = 3k + 1$, $3n + 2$ é par. Note que a sentença "$3n + 2$ é par" é o mesmo que $\neg p$, pois um inteiro é par se e somente se não for ímpar. Como ambas p e $\neg p$ são verdadeiras, temos uma contradição. Isso completa a demonstração por contradição, demonstrando que se $3n + 2$ é ímpar, então n é ímpar. ◀

Note que podemos também demonstrar por contradição que $p \rightarrow q$ é verdadeira, assumindo que p e $\neg q$ são verdadeiras, e mostrando que q deve ser também verdadeira. Isso implica que $\neg q$ e q são ambas verdadeiras, uma contradição. Essa observação nos diz que podemos tornar uma demonstração direta em uma demonstração por contradição.

DEMONSTRAÇÕES DE EQUIVALÊNCIAS Para demonstrar um teorema que é uma sentença bicondicional, ou seja, que é da forma $p \leftrightarrow q$, mostramos que $p \rightarrow q$ e $q \rightarrow p$ são ambas verdadeiras. A validade desse método baseia-se na tautologia

$$(p \leftrightarrow q) \leftrightarrow [(p \rightarrow q) \wedge (q \rightarrow p)].$$

EXEMPLO 12 Prove o teorema "Se n é um inteiro positivo, então n é ímpar se e somente se n^2 for ímpar".

Solução: Esse teorema tem a forma "p se e somente se q", em que p é "n é ímpar" e q é "n^2 é ímpar". (Como usual, não explicitamos a expressão com quantificador universal.) Para demonstrar esse teorema, precisamos mostrar que $p \rightarrow q$ e $q \rightarrow p$ são verdadeiras.

Exemplos Extras

Já mostramos (no Exemplo 1) que $p \rightarrow q$ é verdadeira e (no Exemplo 8) que $q \rightarrow p$ é verdadeira.

Como evidenciamos que ambas $p \rightarrow q$ e $q \rightarrow p$ são verdadeiras, mostramos que o teorema é verdadeiro. ◀

Algumas vezes um teorema determina que muitas proposições são equivalentes. Esses teoremas determinam que as proposições $p_1, p_2, p_3, \ldots, p_n$ são equivalentes. Isso pode ser escrito por

$$p_1 \leftrightarrow p_2 \leftrightarrow \ldots \leftrightarrow p_n,$$

que significa que todas as n proposições têm o mesmo valor-verdade e, conseqüentemente, que para todo i e j com $1 \leq i \leq n$ e $1 \leq j \leq n$, p_i e p_j são equivalentes. Uma maneira de demonstrar essa equivalência mutua é usar a tautologia

$$[p_1 \leftrightarrow p_2 \leftrightarrow \ldots \leftrightarrow p_n] \leftrightarrow [(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_3) \wedge \ldots \wedge (p_n \rightarrow p_1)].$$

Isso mostra que se as sentenças condicionais $p_1 \rightarrow p_2, p_2 \rightarrow p_3, \ldots, p_n \rightarrow p_1$ podem ser demonstradas como verdadeiras, então as proposições $p_1, p_2, \ldots, p_n$ são todas equivalentes.

Isso é muito mais eficiente que demonstrar que $p_i \rightarrow p_j$ para todo $i \neq j$ com $1 \leq i \leq n$ e $1 \leq j \leq n$.

Quando demonstramos que algumas sentenças são equivalentes, podemos estabelecer uma cadeia de sentenças condicionais que escolhermos tão grande quanto possível para trabalhar sobre essa cadeia e ir de qualquer uma dessas sentenças para outra. Por exemplo, podemos evidenciar que p_1, p_2 e p_3 são equivalentes mostrando que $p_1 \rightarrow p_3, p_3 \rightarrow p_2$ e $p_2 \rightarrow p_1$.

EXEMPLO 13 Demonstre que estas sentenças sobre o inteiro n são equivalentes:

p_1: n é par.
p_2: $n - 1$ é ímpar.
p_3: n^2 é par.

Solução: Vamos demonstrar que essas três sentenças são equivalentes mostrando que as condicionais $p_1 \rightarrow p_2, p_2 \rightarrow p_3$, e $p_3 \rightarrow p_1$ são verdadeiras.

Vamos usar uma demonstração direta para mostrar que $p_1 \rightarrow p_2$. Suponha que n é par. Então $n = 2k$ para algum inteiro k. Conseqüentemente, $n - 1 = 2k - 1 = 2(k - 1) + 1$. Isso significa que $n - 1$ é ímpar, pois é da forma $2m + 1$, em que m é o inteiro $k - 1$.

Também vamos usar uma demonstração direta para mostrar que $p_2 \rightarrow p_3$. Agora, suponha que $n - 1$ é ímpar. Então $n - 1 = 2k + 1$ para algum inteiro k. Portanto, $n = 2k + 2$, então, $n^2 = (2k + 2)^2 = 4k^2 + 8k + 4 = 2(2k^2 + 4k + 2)$. Isso significa que n^2 é o dobro do inteiro $2k^2 + 4k + 2$, e, portanto, é par.

Para demonstrar $p_3 \rightarrow p_1$, usaremos uma demonstração por contraposição. Ou seja, provamos que se n não é par, então n^2 não é par. Isso é o mesmo que demonstrar que se n é ímpar, então n^2 é ímpar, o que já demonstramos no Exemplo 1. Isso completa a demonstração. ◀

CONTRA-EXEMPLOS Na Seção 1.3, dissemos que, para demonstrar que uma sentença da forma $\forall x\, P(x)$ é falsa, precisamos apenas encontrar um **contra-exemplo**, que é um exemplo de x para o qual $P(x)$ é falsa. Quando recebemos uma sentença da forma $\forall x\, P(x)$, a qual acreditamos ser falsa ou não conseguimos demonstrar por nenhum método, procuramos por contra-exemplos. Vamos ilustrar o uso de contra-exemplos no Exemplo 14.

EXEMPLO 14 Mostre que a sentença "Todo inteiro positivo é a soma dos quadrados de dois inteiros" é falsa.

Solução: Para mostrar que a sentença é falsa, procuramos um contra-exemplo, que será um inteiro que não é a soma dos quadrados de dois inteiros. Isso não vai demorar muito, pois 3 não pode ser escrito como a soma dos quadrados de dois inteiros. Para mostrar esse caso, note que os quadrados que não excedem 3 são $0^2 = 0$ e $1^2 = 1$. Mais que isso, não há como obter 3 da soma desses dois quadrados, com apenas dois termos. Conseqüentemente, mostramos que "Todo inteiro positivo é a soma dos quadrados de dois inteiros" é falsa. ◀

Erros em Demonstrações

Existem muitos erros comuns na construção de demonstrações matemáticas. Vamos brevemente descrever alguns deles aqui. Os erros mais comuns são erros em aritmética e álgebra básica. Até matemáticos profissionais cometem esses erros, especialmente quando estão trabalhando com fórmulas muito complicadas. Sempre que usarmos essas computações, ou passagens, devemos verificá-las tão cuidadosamente quanto possível. (Você deve também revisar os aspectos problemáticos da álgebra básica, principalmente antes de estudar a Seção 4.1.)

Cada passo de uma demonstração matemática precisa ser correto e a conclusão precisa seguir logicamente os passos que a precedem. Muitos erros resultam da introdução de passos que não seguem logicamente aqueles que o precedem. Isso está ilustrado nos exemplos 15 a 17.

EXEMPLO 15 O que está errado com a famosa suposta "demonstração" de que $1 = 2$?

"Demonstração": Usamos estes passos, em que a e b são dois inteiros positivos iguais

Passo	**Razão**
1. $a = b$	Dado
2. $a^2 = ab$	Multiplicando ambos os membros de (1) por a.
3. $a^2 - b^2 = ab - b^2$	Subtraindo b^2 de ambos os lados de (2)
4. $(a - b)(a + b) = b\,(a - b)$	Fatorando ambos os membros de (3)
5. $a + b = b$	Dividindo ambos os lados de (4) por $a - b$
6. $2b = b$	Substituindo a por b em (5), pois $a = b$, e simplificando
7. $2 = 1$	Dividindo ambos os membros de (6) por b

Solução: Todos os passos são válidos exceto um, o passo 5, onde dividimos ambos os lados por $a - b$. O erro está no fato de $a - b$ ser zero; dividir ambos os lados de uma equação pela mesma quantidade é válido se essa quantidade não é zero. ◀

EXEMPLO 16 O que está errado com a "demonstração"?

"Teorema": Se n^2 é positivo, então n é positivo.

"Demonstração": Suponha que n^2 é positivo. Como a sentença condicional "Se n é positivo, então n^2 é positivo" é verdadeira, podemos concluir que n é positivo.

Solução: Seja $P(n)$ a proposição "n é positivo" e $Q(n)$, "n^2 é positivo". Então nossa hipótese é $Q(n)$. A sentença "*Se* n é positivo, então n^2 é positivo" é a sentença $\forall n(P(n) \rightarrow Q(n))$. Da hipótese $Q(n)$ e da sentença $\forall n(P(n) \rightarrow Q(n))$ não podemos concluir que $P(n)$, pois não estamos usando uma regra válida de inferência. Pelo contrário, este é um exemplo da falácia de afirmação da conclusão. Um contra-exemplo é dado por $n = -1$ para o qual $n^2 = 1$ é positivo, mas n é negativo. ◀

EXEMPLO 17 O que está errado com a "demonstração"?

> "Teorema": Se n não é positivo, então n^2 não é positivo. (Esta é a contrapositiva do "teorema" no Exemplo 16.)

"Demonstração": Suponha que n não é positivo. Como a sentença condicional "Se n é positivo, então n^2 é positivo" é verdadeira, podemos concluir que n^2 não é positivo.

Solução: Sejam $P(n)$ e $Q(n)$ as proposições do Exemplo 16. Então nossa hipótese é $\neg P(n)$ e a sentença "Se n é positivo, então n^2 é positivo" é a sentença $\forall n(P(n) \rightarrow Q(n))$. Da hipótese $\neg P(n)$ e da sentença $\forall n(P(n) \rightarrow Q(n))$ não podemos concluir que $\neg Q(n)$, pois não estamos usando uma regra válida de inferência. Ao contrário, este é um exemplo da falácia de negação da hipótese. Um contra-exemplo é dado por $n = -1$, como no Exemplo 16. ◀

Finalmente, vamos discutir um tipo de erro particularmente desonesto. Muitos argumentos baseiam-se em uma falácia chamada de **carregando a pergunta.** Essa falácia ocorre quando um ou mais passos da demonstração fundamentam-se na sentença que está sendo demonstrada. Em outras palavras, essa falácia aparece quando uma sentença é demonstrada utilizando-se a própria sentença. Esse é o motivo pelo qual essa falácia é também chamada de **raciocínio circular**.

EXEMPLO 18 O argumento seguinte está correto? Ele supostamente mostra que n é um inteiro par sempre que n^2 é um inteiro par.

> Suponha que n^2 é par. Então $n^2 = 2k$ para algum inteiro k. Seja $n = 2l$ para algum inteiro l. Isso mostra que n é par.

Solução: Este argumento é incorreto. A sentença "Seja $n = 2l$ para algum inteiro l" ocorre na demonstração. Nenhum argumento foi dado para demonstrar que n pode ser escrito como $2l$ para algum inteiro l. Este é um raciocínio circular, pois essa sentença é equivalente à sentença que está sendo demonstrada, "n é par". É claro que o resultado está correto; apenas o método de demonstração está errado. ◀

Cometer erros nas demonstrações faz parte do processo de aprendizado. Quando você comete um erro que alguém encontra, você deve analisar cuidadosamente onde está errando e garantir que não cometerá mais o mesmo erro. Matemáticos profissionais também cometem erros em demonstrações. Algumas demonstrações incorretas de importantes resultados enganaram as pessoas por anos antes de os erros serem encontrados.

Só um Começo

Temos agora desenvolvido um arsenal básico de métodos de demonstrações. Na próxima seção, vamos introduzir outros métodos de demonstração importantes. Também vamos introduzir muitas técnicas importantes de demonstração no Capítulo 4, incluindo indução matemática ou indução

matemática, que pode ser usada para demonstrar resultados que valem para todos os inteiros positivos. No Capítulo 5, vamos introduzir a noção de demonstrações combinatórias.

Nesta seção, introduzimos muitos métodos para demonstrar teoremas da forma $\forall x(P(x) \rightarrow Q(x))$, incluindo demonstrações diretas e demonstrações por contraposição. Existem muitos teoremas desse tipo que têm suas demonstrações facilmente construídas pelo método direto através de hipóteses e definições de termos do teorema. No entanto, é freqüentemente difícil demonstrar um teorema sem utilizar uma demonstração por contraposição ou uma demonstração por contradição, ou alguma outra técnica de demonstração. Na Seção 1.7, vamos direcionar estratégias de demonstrações. Vamos descrever várias possibilidades que podem ser usadas para encontrar demonstrações quando o método direto não funciona. Construir demonstrações é uma arte que só pode ser aprendida através da experiência, incluindo escrever demonstrações, ter uma demonstração sua criticada e ler e analisar demonstrações.

Exercícios

1. Use uma demonstração direta para mostrar que a soma de dois números inteiros ímpares é par.

2. Use uma demonstração direta para mostrar que a soma de dois números inteiros pares é par.

3. Mostre que o quadrado de um número par é um número par, usando a demonstração direta.

4. Mostre que o inverso aditivo, ou negativo, de um número par é um número par, usando a demonstração direta.

5. Demonstre que se $m + n$ e $n + p$ são números inteiros pares, em que m, n e p são números inteiros, então $m + p$ é par. Que tipo de demonstração você utilizou?

6. Use uma demonstração direta para mostrar que o produto de dois números ímpares é ímpar.

7. Use uma demonstração direta para mostrar que todo número inteiro ímpar é a diferença de dois quadrados.

8. Demonstre que se n é um quadrado perfeito, então $n + 2$ não é um quadrado perfeito.

9. Use um demonstração por contradição para provar que a soma de um número irracional e um racional é irracional.

10. Use uma demonstração direta para mostrar que o produto de dois números racionais é racional.

11. Demonstre ou contrarie que o produto de dois números irracionais é irracional.

12. Demonstre ou contrarie que o produto de um número racional diferente de zero e um número irracional é irracional.

13. Demonstre que se x é irracional, então $1/x$ é irracional.

14. Demonstre que se x é racional e $x \neq 0$, então $1/x$ é racional.

15. Use uma demonstração por contraposição para mostrar que se $x + y \geq 2$, em que x e y são números reais, então $x \geq 1$ ou $y \geq 1$.

☞ **16.** Demonstre que se m e n são números inteiros e mn é par, então m *é par* ou n é par.

17. Mostre que se n é um número inteiro e $n^3 + 5$ é ímpar, então n é par, usando:

a) uma demonstração por contraposição.

b) uma demonstração por contradição.

18. Demonstre que se n é um número inteiro e $3n + 2$ é par, então n é par, usando:

a) uma demonstração por contraposição.

b) uma demonstração por contradição.

19. Demonstre a proposição $P(0)$, em que $P(n)$ é a proposição "Se n é um número inteiro positivo maior que 1, então $n^2 > n$". Qual tipo de demonstração você utilizou?

20. Demonstre a proposição $P(1)$, em que $P(n)$ é a proposição "Se n é um número inteiro positivo, então $n^2 \geq n$". Qual tipo de demonstração você utilizou?

21. Assuma $P(n)$ como a proposição "Se a e b são números reais positivos, então $(a + b)^n \geq a^n + b^n$". Comprove que $P(1)$ é verdadeira. Qual tipo de demonstração você utilizou?

22. Mostre que se você pegar 3 meias de uma gaveta, com apenas meias azuis e pretas, você deve pegar ou um par de meias azuis ou um par de meias pretas.

23. Mostre que pelo menos 10 de quaisquer 64 dias escolhidos devem cair no mesmo dia da semana.

24. Mostre que pelo menos 3 de quaisquer 25 dias escolhidos devem cair no mesmo mês do ano.

25. Use uma demonstração por contradição para mostrar que não há um número racional r para que $r^3 + r + 1 = 0$. [*Dica:* Assuma que $r = a/b$ seja uma raiz, em que a e b são números inteiros e a/b é o menor termo. Obtenha uma equação que envolva números inteiros, multiplicando-os por b^3. Então, veja se a e b são pares ou ímpares.]

26. Demonstre que se n é um número inteiro positivo, então n é par se e somente se $7n + 4$ for par.

27. Demonstre que se n é um número inteiro positivo, então n é ímpar se e somente se $5n + 6$ for ímpar.

28. Demonstre que $m^2 = n^2$ se e somente se $m = n$ ou $m = -n$.

29. Demonstre ou contrarie que se m e n são números inteiros, tal que $mn = 1$, então ou $m = 1$ e $n = 1$, ou $m = -1$ e $n = -1$.

30. Mostre que essas três proposições são equivalentes, em que a e b são números reais: (*i*) a é menor que b, (*ii*) a média de a e b é maior que a, e (*iii*) a média de a e b é menor que b.

31. Mostre que essas proposições sobre o número inteiro x são equivalentes: (*i*) $3x + 2$ é par, (*ii*) $x + 5$ é ímpar, (*iii*) x^2 é par.

32. Mostre que essas proposições sobre o número real x são equivalentes: (*i*) x é racional, (*ii*) $x/2$ é racional, e (*iii*) $3x - 1$ é racional.

33. Mostre que essas proposições sobre o número real x são equivalentes: (*i*) x é irracional, (*ii*) $3x + 2$ é irracional, (*iii*) $x/2$ é irracional.

34. Esta é a razão para encontrar as soluções da equação $\sqrt{2x^2 - 1} = x$ correta? (1) $\sqrt{2x^2 - 1} = x$ é dado; (2) $2x^2 - 1 = x^2$, obtido pelo quadrado dos dois lados de (1); (3) $x^2 - 1 = 0$, obtido pela subtração de x^2 dos dois lados de (2); (4) $(x - 1)(x + 1) = 0$, obtido pela fatoração do lado esquerdo de $x^2 - 1$; (5) $x = 1$ ou $x = -1$, confirmado, pois $ab = 0$ implica que $a = 0$ ou $b = 0$.

35. Os passos abaixo para encontrar as soluções de $\sqrt{x + 3} = 3 - x$ são corretos? (1) $\sqrt{x + 3} = 3 - x$ é dado; (2) $x + 3 = x^2 - 6x + 9$, obtido tirando a raiz quadrada dos dois lados de (1); (3) $0 = x^2 - 7x + 6$, obtido pela subtração de $x + 3$ dos dois lados de (2); (4) $0 = (x - 1)(x - 6)$, obtido pela fatoração do lado direito de (3); (5) $x = 1$ ou $x = 6$, tirado de (4) porque $ab = 0$ implica que $a = 0$ ou $b = 0$.

36. Comprove que as proposições p_1, p_2, p_3 e p_4 podem ser equivalentes mostrando que $p_1 \leftrightarrow p_4$, $p_2 \leftrightarrow p_3$ e $p_1 \leftrightarrow p_3$.

37. Mostre que as proposições p_1, p_2, p_3, p_4 e p_5 podem ser equivalentes, demonstrando que as proposições condicionais $p_1 \rightarrow p_4$, $p_3 \rightarrow p_1$, $p_4 \rightarrow p_2$, $p_2 \rightarrow p_5$ e $p_5 \rightarrow p_3$ são verdadeiras.

38. Encontre um contra-exemplo para a proposição: todo número inteiro positivo pode ser escrito como a soma dos quadrados de três números inteiros.

39. Comprove que pelo menos um dos números reais $a_1, a_2, \ldots, a_n$ é maior que ou igual ao valor da média desses números. Que tipo de demonstração você utilizou?

40. Use o Exercício 39 para mostrar que se os primeiros 10 números inteiros positivos forem colocados em círculo, em qualquer ordem, haverá três números inteiros, em localização consecutiva no círculo, que terão uma soma maior que ou igual a 17.

41. Comprove que se n é um número inteiro, estas quatro proposições são equivalentes: (*i*) n é par, (*ii*) $n + 1$ é ímpar, (*iii*) $3n + 1$ é ímpar, (*iv*) $3n$ é par.

42. Comprove que estas quatro proposições sobre o número inteiro n são equivalentes: (*i*) n^2 é ímpar, (*ii*) $1 - n$ é par, (*iii*) n^3 é ímpar, (*iv*) $n^2 + 1$ é par.

1.7 Métodos de Demonstração e Estratégia

Introdução

Auto-avaliação

Na Seção 1.6 introduzimos uma variedade de métodos de demonstrações e ilustramos como cada método pode ser usado. Nesta seção vamos continuar neste esforço. Vamos introduzir muitos outros importantes métodos de demonstrações, incluindo demonstrações em que consideramos diferentes casos separadamente e demonstrações em que comprovamos a existência de objetos com determinada propriedade desejada.

Na Seção 1.6 apenas discutimos brevemente a estratégia por trás da construção das demonstrações. Essa estratégia inclui a seleção de um método de demonstrações e então a construção com sucesso de um argumento passo a passo, com base nesse método. Nesta seção, depois que tivermos desenvolvido um grande arsenal de métodos de demonstração, vamos estudar alguns aspectos adicionais da arte e da ciência das demonstrações. Vamos prover avanços em como encontrar demonstrações trabalhando de trás para frente e adaptando demonstrações existentes.

Quando matemáticos trabalham, eles formulam conjecturas e tentam comprová-las ou tentam encontrar um contra-exemplo. Vamos brevemente descrever esse processo aqui, comprovando resultados sobre ladrilhar tabuleiros de xadrez com dominós ou outros tipos de peças. Olhando para esse método de ladrilhar, podemos ser capazes de rapidamente formular conjecturas e comprovar teoremas sem que tenhamos desenvolvido uma teoria.

Vamos concluir a seção discutindo o papel das questões abertas. Em particular, vamos discutir alguns problemas interessantes que apenas foram resolvidos depois de permanecerem abertos por centenas de anos ou porque ainda estão abertos.

Demonstração por Exaustão e Demonstração por Casos

Algumas vezes, não podemos comprovar um teorema usando um único argumento que satisfaça todos os casos possíveis. Vamos agora introduzir um método que pode ser usado para comprovar

teoremas, considerando diferentes casos separadamente. Esse método baseia-se em uma regra de inferência que vamos introduzir agora. Para comprovar uma sentença condicional da forma

$$(p_1 \vee p_2 \vee \ldots \vee p_n) \rightarrow q$$

a tautologia

$$[(p_1 \vee p_2 \vee \ldots \vee p_n) \rightarrow q] \leftrightarrow [(p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \ \wedge \ldots \wedge (p_n \rightarrow q)]$$

pode ser usada como regra de inferência. Isso mostra que o condicional original com a hipótese formada por uma disjunção das proposições $p_1, p_2, \ldots, p_n$ pode ser comprovada verificando-se cada uma das n condicionais $p_i \rightarrow q$, $i = 1, 2, \ldots, n$, individualmente. Esse argumento é chamado de **demonstração por casos**. Algumas vezes para comprovar que uma sentença condicional $p \rightarrow q$ é verdadeira, é conveniente usar a disjunção $p_1 \vee p_2 \vee \ldots \vee p_n$ em vez de p como hipótese da sentença condicional, em que p e $p_1 \vee p_2 \vee \ldots \vee p_n$ são equivalentes.

DEMONSTRAÇÃO POR EXAUSTÃO Alguns teoremas podem ser comprovados examinando-se um número relativamente pequeno de exemplos. Essas demonstrações são chamadas de **demonstrações por exaustão**, pois procedem pela exaustão de todas as possibilidades. Uma demonstração por exaustão é um tipo especial de demonstração por casos em que cada caso envolve apenas a demonstração de um simples exemplo. Vamos ver algumas ilustrações de demonstrações por exaustão.

EXEMPLO 1 Comprove que $(n + 1)^2 \geq 3^n$ se n é um inteiro positivo com $n \leq 4$.

Solução: Vamos usar a demonstração por exaustão. Apenas precisamos verificar que a inequação $(n + 1)^2 \geq 3^n$ é verdadeira quando $n = 1, 2, 3$ e 4. Para $n = 1$, temos $(n + 1)^2 = 2^2 = 4$ e $3^n = 3^1 = 3$; para $n = 2$, temos $(n + 1)^2 = 3^2 = 9$ e $3^n = 3^2 = 9$; para $n = 3$, temos $(n + 1)^3 = 4^3 = 64$ e $3^n = 3^3 = 27$; e para $n = 4$, temos $(n + 1)^3 = 5^3 = 125$ e $3^n = 3^4 = 81$. Em cada um dos quatro casos, vemos que $(n + 1)^2 \geq 3^n$. Usamos o método de demonstração por exaustão para demonstrar que $(n + 1)^2 \geq 3^n$ se n é um número inteiro positivo com $n \leq 4$. ◀

Exemplos Extras

EXEMPLO 2 Demonstre que os únicos inteiros positivos consecutivos não excedendo 100 que são potências perfeitas são 8 e 9. (Um inteiro é uma **potência perfeita** se for igual a n^a, em que a é um inteiro maior que 1.)

Solução: Podemos demonstrar esse fato mostrando que o único par n, $n + 1$ de inteiros positivos consecutivos que são potências perfeitas com $n < 100$ ocorre quando $n = 8$. Podemos demonstrar esse fato examinando os números inteiros positivos n não excedendo 100; primeiro verificamos quando n é uma potência perfeita e, se for, verificamos se $n + 1$ também o é. O método mais rápido para fazer isto é simplesmente olhar para todas as potências perfeitas não excedendo 100. Os quadrados que não excedem 100 são 1, 4, 9, 16, 25, 36, 49, 64, 81 e 100. Os cubos que não excedem 100 são 1, 8, 27 e 64. As quartas potências que não excedem 100 são 1, 16 e 81. As quintas potências que não excedem 100 são 1 e 32. As sextas potências de números inteiros que não excedem 100 são 1 e 64. Não existem outras potências maiores que as sextas que excedam 100, exceto o número 1. Olhando para essa lista de potências, vemos que $n = 8$ é a única potência perfeita para a qual $n + 1$ também é uma potência perfeita. Ou seja, $2^3 = 8$ e $3^2 = 9$ são as únicas duas potências perfeitas consecutivas que não excedem 100. ◀

Podemos recorrer às demonstrações por exaustão apenas quando é necessário verificar um número relativamente pequeno de instâncias da sentença. Computadores não se incomodam quando é pedido para verificar um número muito grande de instâncias, mas eles têm limitações. Note que nenhum computador pode verificar todas as instâncias quando é impossível listar todas as possibilidades.

DEMONSTRAÇÕES POR CASOS Uma demonstração por casos deve cobrir todos as possibilidades que aparecem no teorema. Ilustramos demonstrações por casos com alguns exemplos. Em cada exemplo, você deve verificar que são cobertos todos os casos possíveis.

EXEMPLO 3 Demonstre que se n é um inteiro, então $n^2 \geq n$.

Solução: Podemos demonstrar que $n^2 \geq n$ para todos os inteiros considerando três casos, quando $n = 0$, quando $n \geq 1$ e quando $n \leq -1$. Dividimos a demonstração em três casos, pois é rápido demonstrar o resultado considerando zero, inteiros positivos e inteiros negativos separadamente.

Caso (i). Quando $n = 0$, como $0^2 = 0$, vemos que $0^2 \geq 0$. Disso segue que $n^2 \geq n$ é verdadeira nesse caso.

Exemplos Extras

Caso (ii). Quando $n \geq 1$, quando multiplicamos ambos os membros da inequação $n \geq 1$ pelo número inteiro positivo n, obtemos $n \cdot n \geq n \cdot 1$. Isso implica que $n^2 \geq n$ para $n \geq 1$.

Caso (iii). Nesse caso, $n \leq -1$. No entanto, $n^2 \geq 0$. Disso segue que $n^2 \geq n$.

Como a inequação $n^2 \geq n$ é verdadeira para os três casos, podemos concluir que se n é um inteiro, então $n^2 \geq n$. ◀

EXEMPLO 4 Use uma demonstração por casos para mostrar que $|xy| = |x||y|$, em que x e y são números reais. (Lembre-se de que $|a|$, o valor absoluto de a, é igual a a quando $a \geq 0$ e igual a $-a$ quando $a \leq 0$.)

Solução: Em nossa demonstração desse teorema, vamos remover os valores absolutos usando o fato de que $|a| = a$ quando $a \geq 0$ e $|a| = -a$ quando $a < 0$. Como ambos $|x|$ e $|y|$ ocorrem em nossa fórmula, vamos precisar de quatro casos: *(i)* x e y ambos não negativos, *(ii)* x não negativo e y negativo, *(iii)* x negativo e y não negativo e *(iv)* x negativo e y negativo.

(Note que podemos remover o valor absoluto, fazendo a escolha apropriada dos sinais em cada caso.)

Caso (i). Vemos que $p_1 \rightarrow q$, pois $xy \geq 0$ quando $x \geq 0$ e $y \geq 0$, então $|xy| = xy = |x||y|$.

Caso (ii). Para ver que $p_2 \rightarrow q$, note que se $x \geq 0$ e $y < 0$, então $xy \leq 0$, logo $|xy| = -xy = x(-y) = |x||y|$. (Aqui, como $y < 0$, temos $|y| = -y$.)

Caso (iii). Para ver que $p_3 \rightarrow q$, seguimos o mesmo raciocínio como no caso anterior com os papéis de x e y invertidos.

Caso (iv). Para ver que $p_4 \rightarrow q$, note que quando $x < 0$ e $y < 0$, daí segue que $xy > 0$. Logo, $|xy| = xy = (-x)(-y) = |x||y|$.

Como completamos os quatro casos e esses casos são todas as possibilidades, podemos concluir que $|xy| = |x||y|$, sempre que x e y são números reais. ◀

ALAVANCANDO DEMONSTRAÇÕES POR CASOS Os exemplos que apresentamos para ilustrar demonstrações por casos proveram algum *insight* sobre quando usar esse método de demonstração. Em particular, quando não é possível tratar todos os casos ao mesmo tempo, uma demonstração por casos deve ser considerada. Mas quando usar essa demonstração? Geralmente, tentar uma demonstração por casos quando não existe um meio óbvio de começar a demonstração, mas também quando informações extras de cada caso podem ser usadas para seguir a demonstração. O Exemplo 5 ilustra como o método de demonstrações por casos pode ser usado efetivamente.

EXEMPLO 5 Formule uma conjectura sobre os dígitos decimais que ocorrem como algarismo das unidades nos quadrados de um número inteiro e demonstre seu resultado.

Solução: Os menores quadrados perfeitos são 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, e assim por diante. Notamos que os dígitos que ocorrem com algarismos das unidades dos quadrados são 0, 1, 4, 5, 6 e 9, com 2, 3, 7 e 8 não aparecendo como o último dígito dos qua-

drados. Conjeturamos este teorema: O último dígito decimal de um quadrado perfeito é 0, 1, 4, 5, 6 ou 9. Como podemos demonstrar esse teorema?

Primeiro, note que podemos expressar um inteiro n como $10a + b$, em que a e b são inteiros positivos e b é 0, 1, 2, 3, 4, 5, 6, 7, 8 ou 9. Aqui, a é o inteiro obtido quando subtraímos o algarismo decimal final do n de n e dividimos por 10. Depois, note que $(10a + b)^2 = 100a^2 + 20ab + b^2 = 10(10a^2 + 2b) + b^2$, então o algarismo final de n^2 é o mesmo algarismo final de b^2. Além disso, note-se que o dígito decimal final de b^2 é o mesmo dígito final de $(10 - b)^2 = 100 - 20b + b^2$. Conseqüentemente, podemos reduzir nossa demonstração em seis casos.

Caso (i). O dígito final de n é 1 ou 9. Então o dígito decimal final de n^2 é o dígito decimal final de $1^2 = 1$ ou $9^2 = 81$, ou seja, 1.

Caso (ii). O dígito final de n é 2 ou 8. Então o dígito decimal final de n^2 é o dígito decimal final de $2^2 = 4$ ou $8^2 = 64$, ou seja, 4.

Caso (iii). O dígito final de n é 3 ou 7. Então o dígito decimal final de n^2 é o dígito decimal final de $3^2 = 9$ ou $7^2 = 49$, ou seja, 9.

Caso (iv). O dígito final de n é 4 ou 6. Então o dígito decimal final de n^2 é o dígito decimal final de $4^2 = 16$ ou $6^2 = 36$, ou seja, 6.

Caso (v). O dígito final de n é 5. Então o dígito decimal final de n^2 é o dígito decimal final de $5^2 = 25$, ou seja, 5.

Caso (vi). O dígito final de n é 0. Então o dígito decimal final de n^2 é o dígito decimal final de $0^2 = 0$, ou seja, 0.

Como consideramos todos os seis casos, podemos concluir que o dígito decimal final de n^2 em que n é um número inteiro, é 0, 1, 2, 4, 5, 6 ou 9. ◀

Às vezes, podemos eliminar alguns dos casos, restando poucos exemplos, como o Exemplo 6 ilustra.

EXEMPLO 6 Mostre que não existem soluções inteiras para x e y de $x^2 + 3y^2 = 8$.

Solução: Podemos rapidamente reduzir a demonstração apenas verificando poucos casos, pois $x^2 > 8$ quando $|x| \geq 3$ e $3y^2 > 8$ quando $|y| \geq 2$. Disso restam os casos em que x toma um dos valores -2, -1, 0, 1 ou 2 e y toma um dos valores -1, 0, ou 1. Podemos terminar usando uma demonstração exaustiva. Para economizar com os casos restantes, notamos que x^2 só pode ser 0, 1 ou 4, e os possíveis valores de $3y^2$ são 0 e 3, logo vemos que a maior soma possível para os valores de x^2 e $3y^2$ é 7. Conseqüentemente, é impossível termos $x^2 + 3y^2 = 8$ com x e y inteiros. ◀

SEM PERDA DE GENERALIDADE Na demonstração do Exemplo 4, dispensamos o caso (*iii*), em que $x < 0$ e $y \geq 0$, pois é o mesmo que o caso (*ii*), em que $x \geq 0$ e $y < 0$, com os papéis de x e y invertidos. Para encurtar a demonstração, pudemos demonstrar os casos (*ii*) e (*iii*) juntos, assumindo, **sem perda de generalidade**, que $x \geq 0$ e $y < 0$. Implícito nessa sentença está o fato de que podemos provar o caso com $x < 0$ e $y \geq 0$, usando o mesmo argumento utilizado para o caso com $x \geq 0$ e $y < 0$, mas com as mudanças óbvias. Em geral, quando a frase "sem perda de generalidade" é usada em uma demonstração, queremos dizer que demonstrando um caso do teorema, nenhum argumento adicional é necessário para demonstrar o outro caso especificado. Ou seja, o outro caso segue o mesmo argumento, com as mudanças necessárias. É claro que o uso incorreto desse princípio pode levar a erros desafortunados. Às vezes assume-se que não perderemos a generalidade, mas isso leva à perda de generalidade. Esses erros podem ser cometidos por não levarmos em conta que um caso é substancialmente diferente dos outros. Isso pode levar a uma demonstração incompleta ou, possivelmente, errada. De fato, muitas demonstrações incorretas de famosos teoremas tinham seus erros em argumentos que usavam a idéia de "sem perda de generalidade" para demonstrar casos que não poderiam ser rapidamente demonstrados a partir de casos mais simples.

Vamos agora ilustrar uma demonstração em que "sem perda de generalidade" é usada efetivamente.

EXEMPLO 7 Mostre que $(x + y)^r < x^r + y^r$ quaisquer que sejam x e y reais positivos e r é um número real com $0 < r < 1$.

Solução: Sem perda de generalidade, podemos assumir que $x + y = 1$. [Para ver isso, suponha que tenhamos demonstrado o teorema, assumindo que $x + y = 1$. Suponha que $x + y = t$. Então $(x/t) + (y/t) = 1$, o que implica que $((x/t) + (y/t))^r < (x/t)^r + (y/t)^r$. Multiplicando-se ambos os lados dessa última equação por t^r, mostramos que $(x + y)^r < x^r + y^r$.]

Assumindo que $x + y = 1$, como x e y são positivos, temos $0 < x < 1$ e $0 < y < 1$. Como $0 < r < 1$, segue que $0 < 1 - r < 1$, portanto $x^{1-r} < 1$ e $y^{1-r} < 1$. Isso significa que $x < x^r$ e $y < y^r$. Conseqüentemente, $x^r + y^r > x + y = 1$. Isso significa que $(x + y)^r = 1^r < x^r + y^r$. Isso demonstra o teorema para $x + y = 1$.

Como assumimos $x + y = 1$ sem perda de generalidade, sabemos que $(x + y)^r < x^r + y^r$ quaisquer que sejam x e y reais positivos e r é um número real com $0 < r < 1$. ◀

ERROS COMUNS COM DEMONSTRAÇÕES EXAUSTIVAS E DEMONSTRAÇÕES POR CASOS Um erro comum de raciocínio é tirar conclusões incorretas dos exemplos. Não importa quantos exemplos separados são considerados, um teorema não é demonstrado considerando-se exemplos, a menos que todos os possíveis casos sejam cobertos. O problema de demonstrar um teorema é análogo a mostrar que um programa de computador sempre fornece a saída desejada. Não importa quantos valores de entrada são testados, a menos que todos os valores sejam testados, não podemos concluir que o programa sempre produz a saída correta.

EXEMPLO 8 É verdade que todo número inteiro positivo é a soma de 18 quartas potências de inteiros?

Solução: Para determinar quando n pode ser escrito como a soma de 18 quartas potências de inteiros, devemos começar examinando quando n é a soma de 18 quartas potências de inteiros para o menor inteiro positivo. Como as quatro primeiras quartas potências são 0, 1, 16, 81, . . . , se pudermos selecionar 18 termos desses que adicionados resultam n, então n é a soma de 18 quartas potências. Podemos mostrar que todos os números inteiros positivos até o 78 podem ser escritos como a soma de 18 quartas potências. (Os detalhes são deixados para o leitor.) No entanto, se decidirmos que já é o bastante, podemos chegar a uma conclusão errada. Pois não é verdade que todo inteiro positivo pode ser escrito como a soma de 18 quartas potências de inteiros, uma vez que 79 não pode ser escrito como essa soma (como o leitor pode verificar). ◀

Outro erro comum envolve fazer afirmações não garantidas que levam a demonstrações por casos incorretas em que nem todos os casos são considerados. Isso é ilustrado no Exemplo 9.

EXEMPLO 9 O que está errado com esta "demonstração"?

"Teorema": Se x é um número real, então x^2 é um número real positivo.

"Demonstração": Seja p_1 "x é positivo", seja p_2 "x é negativo" e seja q "x^2 é positivo". Para mostrar que $p_1 \rightarrow q$ é verdadeira, note que quando x é positivo, x^2 é positivo, pois é o produto de dois números positivos, x e x. Para mostrar que $p_2 \rightarrow q$, note que, quando x é negativo, x^2 é positivo, pois é o produto de dois números negativos, x e x. Isso completa a demonstração.

Solução: O problema com a demonstração que demos está no fato de não notarmos o caso $x = 0$. Quando $x = 0$, $x^2 = 0$ não é positivo, portanto o suposto teorema é falso. Se p é "x é um número real", então podemos demonstrar resultados em que p é a hipótese com três casos, p_1, p_2 e p_3, em que p_1 é "x é positivo", p_2 é "x é negativo" e p_3 é "$x = 0$" uma vez que temos a equivalência $p \leftrightarrow p_1 \vee p_2 \vee p_3$. ◀

Demonstrações de Existência

Muitos teoremas são afirmações de que determinados objetos de certo tipo existem. Um teorema desse tipo é uma proposição da forma $\exists x\, P(x)$, em que P é um predicado. Uma demonstração de uma proposição da forma $\exists x\, P(x)$ é chamada de **demonstração de existência**. Existem muitos meios de demonstrar um teorema desse tipo. Às vezes, uma demonstração de existência de $\exists x\, P(x)$ pode ser dada encontrando-se um elemento a tal que $P(a)$ é verdadeira. Essas demonstrações de existência são chamadas de **construtivas**. Também é possível dar uma demonstração de existência que seja **não construtiva**; ou seja, não encontramos um elemento a tal que $P(a)$ é verdadeira, mas podemos provar que $\exists x\, P(x)$ é verdadeira de alguma outra maneira. Um método comum de fornecer uma demonstração de existência não construtiva é usar a demonstração por contradição e mostrar que a negação da quantificação existencial implica uma contradição. O conceito de demonstração construtiva de existência é ilustrado no Exemplo 10 e o conceito de não construtiva, no Exemplo 11.

EXEMPLO 10 **Demonstração de Existência Construtiva** Mostre que existe um inteiro positivo que pode ser escrito como a soma de cubos de duas maneiras diferentes.

Exemplos Extras

Solução: Depois de considerável trabalho (tal como uma procura computacional), encontramos

$$1.729 = 10^3 + 9^3 = 12^3 + 1^3.$$

Como mostramos um inteiro positivo que pode ser escrito como a soma de cubos de dois modos diferentes, está feito. ◀

EXEMPLO 11 **Demonstração de Existência Não Construtiva** Mostre que existem números irracionais x e y, tal que x^y *é* racional.

Solução: De acordo com o Exemplo 10 na Seção 1.6, sabemos que $\sqrt{2}$ é irracional. Considere o número $\sqrt{2}^{\sqrt{2}}$. Se ele é racional, temos dois números irracionais x e y com x^y racional, ou seja, $x = \sqrt{2}$ e $y = \sqrt{2}$. Por outro lado, se $\sqrt{2}^{\sqrt{2}}$ é irracional, então podemos tomar $x = \sqrt{2}^{\sqrt{2}}$ e $y = \sqrt{2}$, logo $x^y = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{(\sqrt{2}\cdot\sqrt{2})} = \sqrt{2}^2 = 2$.

Esta demonstração é um exemplo de demonstração de existência não construtiva, pois não encontramos números irracionais x e y, tal que x^y é racional. No entanto, mostramos que ou o par $x = \sqrt{2}, y = \sqrt{2}$ ou o par $x = \sqrt{2}^{\sqrt{2}}, y = \sqrt{2}$ tem a propriedade desejada, mas não mostramos para qual dos dois pares isso funciona! ◀

Demonstrações não construtivas de existência são delicadas, como o Exemplo 12 ilustra.

EXEMPLO 12 **Chomp** é um jogo para dois jogadores. Nesse jogo, biscoitos são dispostos em uma grade retangular. O biscoito na ponta esquerda é envenenado, como mostra a Figura 1(a). Os dois jogadores fazem movimentos alternadamente, um tem de comer um biscoito restante, junto com todos os biscoitos à direita e/ou abaixo deste (veja a Figura 1(b), por exemplo). O perdedor é o jogador que não tiver escolha e tiver de comer o biscoito envenenado. Perguntamos se um dos dois jogadores pode ter uma estratégia vencedora. Ou seja, pode um jogador sempre fazer movimentos que garantam a ele a vitória?

Links

Solução: Vamos dar uma demonstração de existência não construtiva de uma estratégia vencedora para o primeiro jogador. Ou seja, vamos mostrar que o primeiro jogador tem uma estratégia vencedora sem descrevê-la explicitamente.

Primeiro, note que o jogo tem um fim e não pode terminar empatado porque, com cada movimento, ao menos um biscoito é comido; portanto, depois de $m \times n$, o jogo chega ao fim, em

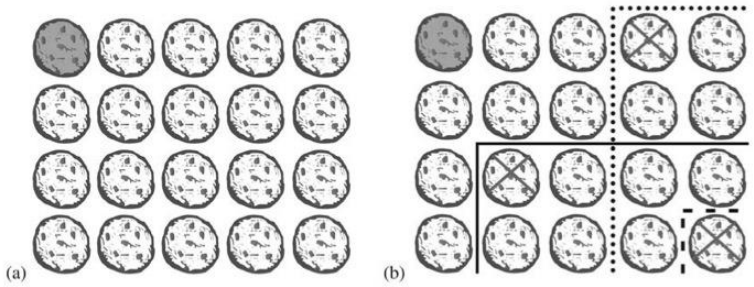


FIGURA 1 (a) Chomp, o Biscoito na Ponta Superior Esquerda é Envenenado (b) Três Movimentos Possíveis

que a grade original é $m \times n$. Agora, suponha que o primeiro jogador comece o jogo comendo exatamente o biscoito no canto direito inferior. Existem duas possibilidades, esse é o primeiro movimento de uma estratégia vencedora para o primeiro jogador ou o segundo jogador pode fazer uma jogada que seja o primeiro movimento de uma estratégia vencedora para o segundo jogador. Nesse segundo caso, em vez de comer o biscoito no canto inferior direito, o primeiro jogador poderia ter feito o mesmo movimento que o segundo jogador fez como seu primeiro movimento de uma estratégia vencedora (e então continuar seguindo aquela estratégia vencedora). Isso garantiria uma vitória para o primeiro jogador.

Note que mostramos que uma estratégia vencedora existe, mas não especificamos uma tal estratégia. Conseqüentemente, a demonstração é não construtiva. De fato, ninguém foi capaz de descrever uma estratégia vencedora para o Chomp que se aplique para todas as grades retangulares, e descreva os movimentos que o primeiro jogador deve seguir. No entanto, estratégias vencedoras podem ser descritas para determinados casos especiais, como quando a grade é quadrada e quando tem apenas duas linhas de biscoitos (veja os exercícios 15 e 16 na Seção 4.2). ◀

Demonstrações de Unicidade

Alguns teoremas afirmam a existência de um único elemento com uma determinada propriedade. Em outras palavras, esses teoremas afirmam que existe exatamente um elemento com essa propriedade. Para demonstrar uma sentença desse tipo, precisamos mostrar que um elemento com essa propriedade existe e que nenhum outro elemento tem essa propriedade. As duas partes de uma **demonstração de unicidade** são:

Existência: Mostramos que um elemento x com a propriedade desejada existe.
Unicidade: Mostramos que se $y \neq x$, então y não tem a propriedade desejada.

Equivalentemente, podemos mostrar que se x e y têm ambos essa propriedade, então $x = y$.

Observação: Mostrar que existe um único elemento x, tal que $P(x)$ é o mesmo que demonstrar a sentença $\exists x(P(x) \wedge \forall y(y \neq x \rightarrow \neg P(y)))$.

Vamos ilustrar os elementos de uma demonstração de unicidade no Exemplo 13.

EXEMPLO 13 Mostre que se a e b são números reais e $a \neq 0$, então existe um único número real r, tal que $ar + b = 0$.

Solução: Primeiro, note que o número real $r = -b/a$ é uma solução para $ar + b = 0$, pois $a(-b/a) + b = -b + b = 0$. Conseqüentemente, um número real r existe para o qual $ar + b = 0$. Esta é a parte de existência da demonstração.

Exemplos Extras

Segundo, suponha que s é um número real, tal que $as + b = 0$. Então, $ar + b = as + b$, em que $r = -b/a$. Subtraindo b de ambos os lados, encontramos $ar = as$. Dividindo ambos os lados dessa última equação por a, que não é zero, vemos $r = s$. Isso significa que se $s \neq r$, então $as + b \neq 0$. Isso estabelece a parte da unicidade da demonstração. ◀

Estratégias de Demonstração

Encontrar demonstrações pode ser um trabalho desafiador. Quando você é confrontado com uma sentença para demonstrar, você deve primeiro substituir os termos por suas definições e, então, cuidadosamente analisar o que as hipóteses e a conclusão significam. Depois disso, você pode tentar demonstrar o resultado usando um dos métodos de demonstração avaliados. Geralmente, se a sentença é uma sentença condicional, você deve primeiro tentar uma demonstração direta; se esta

Links

GODFREY HAROLD HARDY (1877–1947) Hardy, nascido em Cranleigh, Surrey, Inglaterra, foi o mais velho de dois filhos de Isaac Hardy e Sophia Hall Hardy. Seu pai era mestre em geografia e desenho pela Escola de Cranleigh e também dava aulas de canto e jogava futebol. Sua mãe dava aulas de piano e mantinha uma pensão para jovens estudantes. Os pais de Hardy eram devotos da educação de seus filhos. Hardy demonstrou suas habilidades com números com apenas dois anos de idade, quando começou a desenhar números em milhões. Ele teve um tutor em matemática em vez de freqüentar a escola regular. Foi para a Winchester College, escola de segundo grau particular, aos 13 anos ao ganhar uma bolsa de estudos. Ele era um excelente aluno e demonstrou um intenso interesse por matemática. Hardy entrou para o Trinity College, Cambridge, em 1896, com uma bolsa de estudos e ganhou vários prêmios durante o tempo que passou por lá, graduando-se em 1899.

Hardy manteve o cargo de professor em matemática na Trinity College, na Cambridge University, de 1906 a 1919, quando foi indicado para a cadeira de geometria de Sullivan, em Oxford. Ele ficou descontente com Cambridge por causa da demissão do famoso filósofo e matemático Bertrand Russell, do Trinity, pelas atividades antibélicas, e não apoiava algumas atitudes administrativas tomadas. Em 1931, ele retornou a Cambridge como professor de matemática pura, onde permaneceu até sua aposentadoria em 1942. Ele foi um matemático purista e manteve uma visão elitista da matemática, esperando que sua pesquisa nunca fosse aplicada.

Ironicamente, Hardy é talvez o mais conhecido dos criadores da lei Hardy–Weinberg, que predita os modelos de herança. Seu trabalho nessa área aparece como uma carta no periódico *Science,* no qual ele usa as idéias de álgebra simples para demonstrar os erros em um artigo sobre genética. Hardy trabalhou primeiramente com a teoria de números e função, explorando alguns tópicos como a função zeta de Riemann, séries de Fourier e a distribuição dos números primos. Ele fez muitas contribuições importantes a destacados problemas, como os de Waring sobre a representação dos números inteiros positivos como soma da k-ésima potência e o problema da representação dos números inteiros ímpares como a soma de três primos. Hardy é também lembrado pelas suas colaborações com John E. Littlewood, colega de trabalho com quem escreveu mais de 100 artigos, e com o famoso matemático indiano prodígio Srinivasa Ramanujan. Sua colaboração com Littlewood formou a piada de que existiam apenas três matemáticos importantes naquela época, Hardy, Littlewood e Hardy–Littlewood, embora algumas pessoas pensassem que Hardy teria inventado uma pessoa fictícia, Littlewood, porque Littlewood era muito pouco visto fora de Cambridge. Hardy teve a sabedoria de reconhecer a genialidade de Ramanujan a partir de escritos não convencionais, mas extremamente criativos enviados por Ramanujan para ele, enquanto outros matemáticos falharam nesta tarefa. Hardy trouxe Ramanujan para Cambridge, e ele colaborou em importantes artigos, estabelecendo novos resultados no número de partições de números inteiros. Hardy se interessava pela educação matemática, e seu livro *Um Curso de Matemática Pura* teve um grande efeito na educação primária e secundária em matemática na primeira metade do século XX. Hardy também escreveu *Uma Apologia de um Matemático*, em que ele deu sua resposta à pergunta de que se vale a pena devotar a vida ao estudo da matemática. Neste livro é apresentada a visão dele do que é matemática e o que ela faz.

Hardy teve um grande interesse por esportes. Ele era um fã ávido de críquete e acompanhava os resultados de perto. Em uma determinada partida em que ele estava, tiraram uma foto dele da qual não gostou (apenas cinco fotos são conhecidas), além disso, não gostava de espelhos, cobrindo-os com uma toalha imediatamente ao entrar em um quarto de hotel.

NOTA HISTÓRICA O matemática inglês G. H. Hardy, quando visitava o indiano prodígio Ramanujan no hospital, contou-lhe que se lembrara do número 1.729, o número de táxis que tomara, que era bastante estúpido. Ramanujan respondeu: "Não; é um número muito interessante; é o menor número que pode ser expresso como a soma dos cubos de duas formas diferentes".

falhar, você pode tentar uma demonstração indireta. Se nenhuma dessas tentativas funcionar, você deve tentar uma demonstração por contradição.

RACIOCÍNIO DIRETO E PARA TRÁS Qualquer que seja o método escolhido, você precisa de um ponto de partida para sua demonstração. Para começar uma demonstração direta de uma sentença condicional, comece com as premissas. Usando estas premissas, juntamente com os axiomas e conhecendo teoremas, você pode construir uma demonstração usando uma seqüência de passos que nos leva à conclusão. Esse tipo de raciocínio, chamado de *raciocínio direto*, é o tipo mais comum de raciocínio usado para demonstrar resultados relativamente simples. Similarmente, com raciocínios indiretos, você pode começar com a negação da conclusão e, usando uma seqüência de passos, obter a negação das premissas.

Infelizmente, raciocínio direto é com freqüência difícil de usar para demonstrar resultados mais complicados, pois os raciocínios necessários para alcançar a conclusão desejada podem estar distantes do óbvio. Nesses casos pode ser interessante usar um *raciocínio para trás*. Raciocinando de trás para frente para demonstrar uma sentença q, achamos uma sentença p que pode demonstrar a propriedade que $p \rightarrow q$. (Note que isso não ajuda a encontrar uma sentença r que você pode demonstrar que $q \rightarrow r$, pois esta é uma falácia de carregar a pergunta para concluir de $q \rightarrow r$ e r que q é verdadeira.) Raciocínios de trás para frente serão ilustrados nos exemplos 14 e 15.

Links

SRINIVASA RAMANUJAN (1887–1920) Famoso prodígio matemático, Ramanujan nasceu e foi criado no sul da Índia, próximo à cidade de Madras (agora chamada de Chennai). Seu pai era balconista em uma loja de roupas. Sua mãe contribuía com a renda familiar cantando em um templo local. Ramanujan estudou em uma escola de língua inglesa, expondo seu talento e interesse pela matemática. Com 13 anos, ele usava um livro de teoria para universitários. Quando ele tinha 15 anos, um universitário deu-lhe uma cópia de *Sinopse da Matemática Pura*. Ramanujan decidiu trabalhar com os mais de 6 mil resultados do livro, declarados sem demonstração ou explicação, escrevendo tudo em folhas depois coletadas em um livro de anotações. Terminou o colegial em 1904 e ganhou uma bolsa de estudos para a Universidade de Madras. Inscrito em um currículo de artes finas, ele não freqüentava suas aulas e assistia às aulas de matemática, o que o fez perder sua bolsa. Ele não passou nos exames na universidade quatro vezes, de 1904 a 1907, indo bem apenas em matemática. Durando esse tempo, ele completava seu caderno de anotações com escritos originais, algumas vezes redescobrindo trabalhos já publicados e, em outras, fazendo novas descobertas.

Sem um diploma universitário, era difícil para Ramanujan conseguir um emprego decente. Para sobreviver, ele dependia de ajuda de amigos. Ele foi tutor de estudantes em matemática, mas suas maneiras não convencionais de pensar e sua falta de conhecimento do programa de ensino causavam-lhe problemas. Ele casou-se em 1909, um casamento arranjado com uma garota 9 anos mais nova que ele. Por precisar sustentar a nova família, mudou-se para Madras e foi atrás de um emprego. Ele mostrava suas anotações matemáticas aos empregadores em potencial, mas elas os transtornavam. Entretanto, um professor da Presidency College reconheceu sua genialidade e o apoiou e, em 1912, ele conseguiu um emprego como gerente de contas, ganhando um pequeno salário.

Ramanujan continuou seu trabalho matemático durante esse tempo e publicou seu primeiro artigo em 1910 em um periódico indiano. Ele percebeu que seu trabalho ía além daquilo já feito pelos matemáticos indianos e decidiu escrever aos matemáticos ingleses. O primeiro matemático a quem escreveu, recusou seu pedido de ajuda. Mas, em janeiro de 1913, ele escreveu a G. H. Hardy, que estava inclinado a recusar Ramanujan, mas as proposições matemáticas em sua carta, sem demonstrações, chamaram a atenção de Hardy. Ele decidiu examiná-las de perto com a ajuda de seu colega e colaborador J. E. Littlewood. Eles decidiram, depois de um estudo minucioso, que Ramanujan era provavelmente um gênio, porque suas proposições "poderiam ser escritas apenas por um matemático de alto nível; elas devem ser verdadeiras, porque se forem falsas, ninguém poderia ter a imaginação de inventá-las".

Hardy conseguiu uma bolsa de estudos para Ramanujan, trazendo-o para a Inglaterra, em 1914. Hardy pessoalmente o auxiliou na análise matemática e eles trabalharam juntos por 5 anos, provando teoremas significativos sobre o número de partições de inteiros. Durante esse tempo, Ramanujan fez importantes contribuições para a teoria dos números e também trabalhou com frações, séries infinitas e funções elípticas. Ramanujan teve várias idéias que envolviam determinados tipos de funções e séries, mas suas suposições de teoremas de números primos estavam geralmente erradas. Isso ilustra sua vaga idéia do que constitui uma demonstração correta. Ele foi um dos membros mais novos que se juntou à Irmandade da Sociedade Real. Infelizmente, em 1917, Ramanujan ficou muito doente. Nesta época, pensou-se que ele estava tendo problemas com o clima inglês e tivesse contraído tuberculose. Sabe-se agora que ele sofria de deficiência de uma vitamina, uma vez que Ramanujan era vegetariano. Ele retornou para a Índia em 1919, continuando seus estudos em matemática mesmo quando estava confinado em sua cama. Ele era religioso e acreditava que seu talento matemático procedia da divindade de sua família, Namagiri. Ramanujan considerava a matemática e a religião como que entrelaçadas. Ele dizia que "uma equação para mim não tem significado a menos que expresse um pensamento de Deus". Sua curta vida veio ao final em abril de 1920, quando ele tinha 32 anos de idade. Ramanujan deixou muitas anotações com resultados não publicados. Os escritos nesses cadernos mostram os pensamentos de Ramanujan, mas são limitados. Muitos matemáticos devotaram anos de estudo para explicar e justificar os resultados dos cadernos.

EXEMPLO 14 Dados dois números reais positivos x e y, sua **média aritmética** é $(x + y)/2$ e sua **média geométrica** é $\sqrt{xy}$. Quando comparamos essas médias para pares de números reais positivos distintos, vemos que a média aritmética é sempre maior que a geométrica. [Por exemplo, quando $x = 4$ e $y = 6$, temos $5 = (4 + 6)/2 > \sqrt{4 \cdot 6} = \sqrt{24}$.] Podemos demonstrar que essa inequação é sempre verdadeira?

Solução: Para demonstrar que $(x + y)/2 > \sqrt{xy}$ quando x e y são números reais distintos, podemos pensar de trás para frente. Construímos a seqüência de inequações equivalentes. As inequações equivalentes são:

Exemplos Extras

$$\begin{aligned} (x + y)/2 &> \sqrt{xy}, \\ (x + y)^2/4 &> xy, \\ (x + y)^2 &> 4xy, \\ x^2 + 2xy + y^2 &> 4xy, \\ x^2 - 2xy + y^2 &> 0, \\ (x - y)^2 &> 0. \end{aligned}$$

Como $(x - y)^2 > 0$ quando $x \neq y$, segue que a inequação final é verdadeira. Como todas essas inequações são equivalentes, segue que $(x + y)/2 > \sqrt{xy}$ quando $x \neq y$. Uma vez que fizemos esse raciocínio, podemos facilmente reverter os passos para construir uma demonstração usando um raciocínio direto. Daremos agora esta demonstração.

Suponha que x e y são números reais distintos. Então $(x - y)^2 > 0$, pois o quadrado de um número diferente de zero é positivo (veja Apêndice 1). Como $(x - y)^2 = x^2 - 2xy + y^2$, isso implica que $x^2 - 2xy + y^2 > 0$. Adicionando $4xy$ em ambos os lados, obtemos $x^2 + 2xy + y^2 > 4xy$. Como $x^2 + 2xy + y^2 = (x + y)^2$, isso significa que $(x + y)^2 \geq 4xy$. Dividindo ambos os membros dessa equação por 4, vemos que $(x + y)^2/4 > xy$. Finalmente, tomando raízes quadradas dos dois lados (o que preserva a inequação, pois ambos os lados são positivos), temos $(x + y)/2 > \sqrt{xy}$. Concluímos que se x e y são números reais positivos, então sua média aritmética $(x + y)/2$ é maior que sua média geométrica $\sqrt{xy}$. ◀

EXEMPLO 15 Suponha que duas pessoas participem de um jogo, e que cada um na sua vez toma uma, duas ou três pedras de uma pilha que contém 15 pedras. A pessoa que remove a última pedra ganha o jogo. Mostre que o primeiro jogador pode ganhar, não importando o que o segundo jogador faça.

Solução: Para comprovar que o primeiro jogador pode sempre ganhar o jogo, podemos pensar de trás para frente. No último passo, o primeiro jogador pode ganhar se sobrar uma pilha com uma, duas ou três pedras. O segundo jogador será forçado a deixar uma, duas ou três pedras se ele tiver de jogar com uma pilha que contém quatro pedras. Conseqüentemente, uma maneira de o primeiro jogador ganhar é deixar quatro pedras para o segundo jogador na sua penúltima jogada. O primeiro jogador pode deixar quatro pedras quando existirem cinco, seis ou sete na pilha que o segundo jogador deixou, o que acontecerá quando o segundo jogador fizer seu movimento em uma pilha com oito pedras. Conseqüentemente, para forçar o segundo jogador a deixar cinco, seis ou sete pedras, o primeiro jogador deve deixar oito pedras para o segundo na sua antipenúltima jogada. Isso significa que deve ter nove, dez ou onze pedras para esta jogada. Similarmente, o segundo jogador deve deixar doze pedras quando fizer sua primeira jogada. Podemos reverter este argumento para mostrar que o primeiro jogador deve sempre fazer jogadas para ganhar o jogo, qualquer que sejam as jogadas do segundo. Essas jogadas sucessivamente deixam doze, oito e quatro pedras para o segundo. ◀

ADAPTANDO DEMONSTRAÇÕES DE EXISTÊNCIA Uma excelente maneira de procurar por possíveis métodos que podem ser usados para demonstrar uma sentença é tomar vantagem das demonstrações de existência. Freqüentemente, uma demonstração de existência pode ser adaptada para demonstrar um novo resultado. Mesmo quando esse não é o caso, alguma idéia da demonstração de existência pode ser usada para ajudar. Como demonstrações de existência nos dão dicas para novas demonstrações, você deve ler e entender essas demonstrações quando encontrá-las em seus estudos. Esse processo é ilustrado no Exemplo 16.

EXEMPLO 16 No Exemplo 10 da Secção 1.6, demonstramos que $\sqrt{2}$ é irracional. Agora, conjecturamos que $\sqrt{3}$ é irracional. Podemos adaptar a demonstração do Exemplo 10 da Seção 1.6 para mostrar que $\sqrt{3}$ é irracional?

Exemplos Extras

Solução: Para adaptar a demonstração do Exemplo 10 da Seção 1.6, começamos imitando os passos da demonstração, mas substituindo $\sqrt{2}$ por $\sqrt{3}$. Primeiro, supomos que $\sqrt{3} = d/c$ em que a fração c/d é irredutível. Elevando ambos os lados ao quadrado, temos $3 = c^2/d^2$, logo $3d^2 = c^2$. Podemos usar essa equação para mostrar que 3 deve ser um fator de c e d, similar a como usamos a equação $2b^2 = a^2$ no Exemplo 10 da Seção 1.6 para mostrar que 2 deveria ser um fator de a e b? (Lembre-se de que um inteiro s é um fator de um inteiro t se t/s for um inteiro. Um inteiro n é par se e somente se 2 for um fator de n.) Sim, podemos, mas precisamos de mais alguma munição da teoria dos números, que vamos desenvolver no Capítulo 3. Vamos terminar a demonstração, mas deixaremos as justificativas desses passos para o Capítulo 3. Como 3 é um fator de c^2, este deve ser um fator de c. Mais que isso, como 3 é um fator de c, 9 deve ser um fator de c^2, o que significa que 9 é um fator de $3d^2$. Isso implica que 3 é um fator de d^2, o que significa que 3 é um fator de d. Isso mostra que 3 é um fator de c e d, o que é uma contradição. Depois de termos certeza das justificativas desses passos, teremos mostrado que $\sqrt{3}$ é irracional pela adaptação da demonstração de que $\sqrt{2}$ é irracional. Note que esta demonstração pode ser estendida para mostrar que $\sqrt{n}$ é irracional sempre que n for um inteiro positivo que não é um quadrado perfeito. Deixaremos estes detalhes para o Capítulo 3. ◀

Uma boa dica é procurar demonstrações de existência que você pode adaptar quando for confrontado com a demonstração de um novo teorema, particularmente quando o novo teorema parece semelhante a um que você já demonstrou.

Procurando Contra-exemplos

Na Seção 1.5, introduzimos o uso de contra-exemplos para mostrar que determinadas sentenças são falsas. Quando confrontado com uma conjectura, você deve primeiro tentar demonstrar essa conjectura, e, se suas tentativas não tiverem sucesso, você deve tentar encontrar um contra-exemplo. Se você não conseguir um contra-exemplo, deve tentar demonstrar a sentença. De qualquer forma, procurar contra-exemplos é extremamente importante, pois freqüentemente nos dá algum *insight* sobre o problema. Vamos ilustrar o papel dos contra-exemplos com alguns exemplos.

EXEMPLO 17 No Exemplo 14 da Seção 1.6, mostramos que a sentença "Todo inteiro positivo é a soma de dois quadrados de inteiros" é falsa, encontrando um contra-exemplo. Ou seja, existem inteiros positivos que não podem ser escritos como a soma de dois quadrados de inteiros. Portanto, não podemos escrever todos os inteiros como a soma de quadrados de dois números inteiros, mas talvez possamos escrever todo número inteiro como a soma dos quadrados de três números inteiros. Ou seja, a sentença "Todo número inteiro positivo pode ser escrito como a soma dos quadrados de três números inteiros" é verdadeira ou falsa?

Exemplos Extras

Solução: Como sabemos mostrar que nem todo número inteiro positivo pode ser escrito como a soma de dois quadrados de inteiros, devemos inicialmente ser céticos sobre todo inteiro poder ser escrito como a soma de três quadrados de inteiros. Portanto, vamos primeiro procurar por um contra-exemplo. Ou seja, podemos mostrar que a sentença "Todo número inteiro positivo pode ser

escrito como a soma dos quadrados de três números inteiros" é falsa se encontrarmos um inteiro que não seja a soma de quadrados de três inteiros. Para encontrar um contra-exemplo, tentamos escrever a sucessão dos inteiros como a soma de três quadrados. Vemos que $1 = 0^2 + 0^2 + 1^2$, $2 = 0^2 + 1^2 + 1^2$, $3 = 1^2 + 1^2 + 1^2$, $4 = 0^2 + 0^2 + 2^2$, $5 = 0^2 + 1^2 + 2^2$, $6 = 1^2 + 1^2 + 2^2$, mas não podemos encontrar um modo de escrever 7 como a soma de três quadrados. Para mostrar que não existem três quadrados que somados resultem 7, notemos que os únicos quadrados possíveis que podemos usar menores que 7 são 0, 1 e 4. Como não existe soma de três termos que resulte 7 com essas parcelas, segue-se que 7 é um contra-exemplo. Concluímos que a sentença "Todo número inteiro positivo pode ser escrito como a soma dos quadrados de três números inteiros" é falsa.

Mostramos que nem todo inteiro positivo pode ser escrito como a soma de quadrados de três inteiros. A próxima pergunta deve ser se podemos escrever todos os inteiros positivos como a soma de quatro quadrados de inteiros. Alguma experimentação mostra evidências de que sim. Por exemplo, $7 = 1^2 + 1^2 + 1^2 + 2^2$, $25 = 4^2 + 2^2 + 2^2 + 1^2$ e $87 = 9^2 + 2^2 + 1^2 + 1^2$. Isso nos faz conjecturar que "Todo inteiro positivo é a soma dos quadrados de quatro inteiros" é verdadeira. Para uma demonstração, veja [Ro05]. ◀

Estratégia de Demonstração em Ação

A matemática é geralmente concebida como se fatos matemáticos estivessem cravados em pedras. Textos de matemática (incluindo este livro) apresentam formalmente teoremas e suas demonstrações. Essas apresentações não nos levam a descobrir o processo matemático. Esse processo começa com a exploração de conceitos e exemplos, fazendo perguntas, formulando conjecturas e tentando valorar essas conjecturas com uma demonstração ou com contra-exemplos. Essas são as atividades do dia-a-dia de um matemático. Acredite ou não, o material apresentado em livros é originalmente concebido dessa forma.

Pessoas formulam conjecturas com base em muitos tipos de evidência. O exame de casos especiais pode nos levar a uma conjectura, como a identificação de possíveis modelos. Alterando hipóteses e conclusões de teoremas conhecidos também podemos ser levados a conjecturas plausíveis. Em outros tempos, conjecturas eram feitas com base na intuição, na crença de que um resultado era verdadeiro. Independentemente de como uma conjectura foi feita, uma vez formulada, o objetivo é demonstrar que é verdadeira ou falsa. Quando um matemático acredita que uma conjectura deve ser verdadeira, ele tenta encontrar uma demonstração. Se não consegue encontrá-la, deve procurar por um contra-exemplo. Quando não encontra um contra-exemplo, deve tornar atrás e tentar provar a conjectura novamente. Embora muitas conjecturas sejam verificadas rapidamente, algumas resistem por centenas de anos e levam ao desenvolvimento de novas partes da matemática. Vamos mencionar algumas famosas conjecturas mais tarde, nesta seção.

Ladrilhando

Links

Podemos ilustrar os aspectos de estratégia de demonstrações através de um breve estudo sobre como fazer coberturas de um tabuleiro de xadrez, como se colocássemos ladrilhos em um piso. Olhando para esse processos, podemos rapidamente descobrir e demonstrar muitos resultados diferentes, usando vários métodos de demonstração. Existe um sem-número de conjecturas que pode ser feito e estudado nessa área. Para começar, precisamos definir alguns termos. Um **tabuleiro de xadrez** é um retângulo dividido em quadrados de mesmo tamanho em linhas horizontais e verticais. O jogo de xadrez é jogado em um tabuleiro com 8 linhas e 8 colunas; esse tabuleiro será chamado de **tabuleiro standard** e é mostrado na Figura 2. Nesta seção, usaremos o termo **tabuleiro** para representar qualquer quadriculado de qualquer tamanho, como, por exemplo, as partes do tabuleiro de xadrez obtidas pela retirada de um ou mais quadrados. Um **dominó** é uma peça retangular formada por dois quadrados, como mostra a Figura 3. Dizemos que um tabuleiro está **ladrilhado** por dominós quando todos os seus quadrados estão cobertos sem haver dominós sobrepostos nem dominós com partes para fora do tabuleiro. Agora podemos desenvolver alguns resultados sobre ladrilhamento de tabuleiros usando dominós.

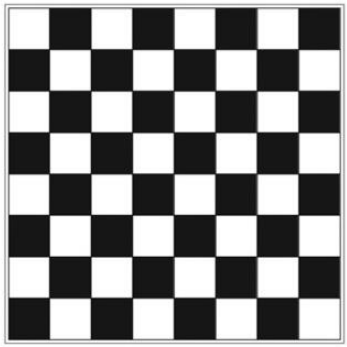


FIGURA 2 O Tabuleiro Standard.

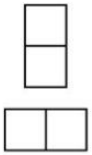


FIGURA 3
Dois Dominós.

EXEMPLO 18 Podemos ladrilhar um tabuleiro standard usando dominós?

Solução: Podemos encontrar muitas maneiras de ladrilhar um tabuleiro standard usando dominós. Por exemplo, podemos ladrilhá-lo colocando 32 dominós horizontais, como mostra a Figura 4. A existência de uma tal situação completa a demonstração construtiva de existência. É claro que existe um grande número de maneiras para fazer essa cobertura. Podemos colocar 32 dominós verticais no tabuleiro ou podemos colocar alguns verticais e outros horizontais. Contudo, para uma demonstração de existência construtiva, precisamos encontrar apenas uma tal situação. ◀

EXEMPLO 19 Podemos ladrilhar um tabuleiro obtido pela remoção de um dos cantos de um tabuleiro standard?

Exemplos Extras

Solução: Para responder a essa questão, note que um tabuleiro standard tem 64 quadrados, portanto, removendo um quadrado, teremos um tabuleiro com 63 quadrados. Agora, suponha que possamos ladrilhar o tabuleiro obtido. O tabuleiro deve ter um número par de quadrados, pois cada dominó cobre dois quadrados e não existem dominós sobrepostos nem que ultrapassam as bordas do tabuleiro. Conseqüentemente, demonstramos por contradição que um tabuleiro de xadrez com um de seus quadrados removidos não pode ser ladrilhado usando dominós, pois esse tabuleiro tem um número ímpar de quadrados. ◀

Agora considere uma situação mais ardilosa.

EXEMPLO 20 Podemos ladrilhar um tabuleiro obtido pela retirada do quadrado superior esquerdo e do inferior direito de um tabuleiro standard, mostrado na Figura 5?

Solução: Um tabuleiro obtido pela retirada de dois quadrados de um tabuleiro standard contém $64 - 2 = 62$ quadrados. Como 62 é par, não podemos dizer imediatamente que não existe essa situação, como fizemos no Exemplo 19, em que demonstramos que não existia um ladrilhamento de um tabuleiro com um quadrado removido. Procurar construir um ladrilhamento desse tabuleiro através da colocação de sucessivos dominós pode ser uma primeira tentativa, como o leitor pode experimentar. No entanto, mesmo depois de muitas tentativas, não conseguimos encontrar essa solução. Como nossos esforços não resultaram em uma solução, somos levados a conjecturar que não existe essa solução.

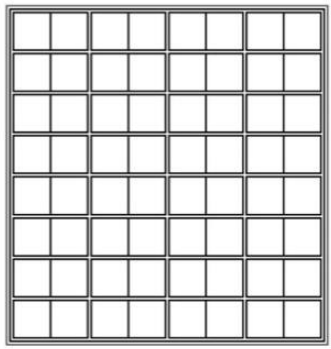


FIGURA 4 Ladrilhando um Tabuleiro Standard.

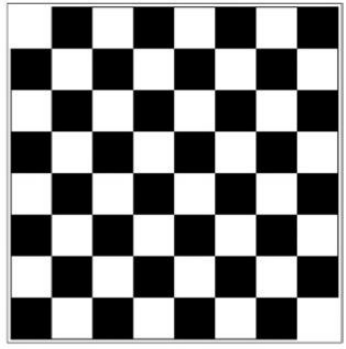


FIGURA 5 O Tabuleiro Standard sem os Dois Cantos Opostos.

Tentamos então demonstrar que não existe ladrilhamento, mostrando que alcançamos uma situação sem saída toda vez que colocamos sucessivamente dominós no tabuleiro. Para construir essa demonstração, devemos considerar todos os possíveis casos que aparecem como se estivéssemos passando por todas as escolhas possíveis na colocação de sucessivos dominós. Por exemplo, temos duas escolhas para cobrir o quadrado na segunda coluna da primeira linha, próximo ao quadrado removido do canto esquerdo superior. Podemos cobri-lo com um dominó na horizontal ou na vertical. Cada uma das escolhas nos leva a muitas escolhas, e assim sucessivamente. Não demorará muito para ver que esse não é um plano interessante para uma pessoa, embora um computador possa ser usado para completar essa demonstração por exaustão. (O Exercício 21 pede para fornecer uma demonstração de que um tabuleiro 4 por 4 com cantos opostos excluídos não pode ser ladrilhado.)

Precisamos de um outro método. Talvez exista uma maneira mais fácil de demonstrar que não existe essa solução para um tabuleiro standard com dois cantos opostos removidos. Como em muitas demonstrações, alguma observação pode ajudar. Vamos colorir os quadrados desse tabuleiro usando branco e preto alternadamente, como na Figura 2. Observe que um dominó cobre sempre um quadrado branco e um preto. Agora note que esse tabuleiro tem um número diferente de quadrados brancos e pretos. Podemos usar essa observação para mostrar por contradição que um tabuleiro standard com cantos opostos removidos não pode ser ladrilhado usando dominós. Agora apresentamos essa demonstração.

Demonstração: Suponha que usemos dominós para ladrilhar um tabuleiro standard com cantos opostos removidos. Note que o tabuleiro standard com cantos opostos removidos contém $64 - 2 = 62$ quadrados. O ladrilhamento usará $62/2 = 31$ dominós. Note que cada dominó cobre um quadrado branco e um preto. Conseqüentemente, o ladrilhamento cobre 31 quadrados brancos e 31 pretos. No entanto, quando removemos dois cantos opostos, 32 dos quadrados restantes são brancos e 30 são pretos ou 32 pretos e 30 brancos. Isso contradiz o fato assumido de que podemos usar dominós para cobrir esse tabuleiro, completando a demonstração. ◀

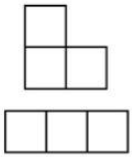


FIGURA 6 Um Triominó à Direita e Um Triominó Reto.

Podemos usar outros tipos de peças além de dominós para ladrilhar. Em vez de dominós, podemos estudar ladrilhamentos que usam peças construídas a partir de quadrados congruentes que são conectados pelos seus lados. Essas peças são os chamados **poliominós**, termo cunhado em 1953 pelo matemático Solomon Golomb, autor de um livro de entretenimento sobre eles [Go94]. Vamos considerar dois poliominós com o mesmo número de quadrados que podemos rotacionar e/ou deslizar para obter outro poliominó. Por exemplo, existem dois tipos de triominós (veja a Figura 6), que são poliominós feitos com três quadrados conectados pelos seus lados. Um

tipo de triominó, o **triominó reto**, tem três quadrados conectados horizontalmente, o outro tipo, o **triominó à direita**, lembra uma letra L, e pode ser rotacionado e/ou deslocado, se necessário. Podemos estudar esses ladrilhamentos de tabuleiros por triominós retos agora; e vamos estudar os ladrilhamentos por triominós à direita na Seção 4.1.

EXEMPLO 21 Podemos usar triominós retos para ladrilhar um tabuleiro standard?

Solução: O tabuleiro standard contém 64 quadrados e cada triominó cobre três quadrados. Conseqüentemente, se triominós ladrilham um tabuleiro, o número de quadrados do tabuleiro deve ser múltiplo de 3. Como 64 não é um múltiplo de 3, triominós não podem ser usados para cobrir um tabuleiro 8 por 8. ◀

No Exemplo 22, consideraremos o problema de ladrilhar um tabuleiro standard sem um canto com triominós retos.

EXEMPLO 22 Podemos usar triominós retos para cobrir um tabuleiro standard sem um de seus cantos? Um tabuleiro 8 por 8 com um canto removido tem $64 - 1 = 63$ quadrados. Qualquer possibilidade de cobrir com triominós retos usará $63/3 = 21$ triominós. No entanto, quando tentamos, não conseguimos encontrar uma solução. Uma demonstração por exaustão não parece promissora. Podemos adaptar nossa demonstração do Exemplo 20 para demonstrar que esse ladrilhamento não existe?

Solução: Vamos colorir os quadrados desse tabuleiro em uma tentativa de adaptar a demonstração por contradição que demos no Exemplo 20, da impossibilidade de ladrilhar, usando dominós para cobrir um tabuleiro com cantos opostos removidos. Como estamos usando triominós retos em vez de dominós, vamos colorir os quadrados usando três cores em vez de duas, como mostra a Figura 7. Note que existem 21 quadrados cinza, 21 pretos e 22 brancos nessa coloração. Depois, faremos uma observação crucial que, quando o triominó cobre três quadrados do tabuleiro, ele cobre um quadrado cinza, um preto e um branco. Depois, note que cada uma das três cores aparece em algum canto do tabuleiro. Em seguida, sem perda de generalidade, podemos assumir que o quadrado retirado é cinza. Portanto, assumimos que sobram 20 quadrados cinza, 21 pretos e 22 brancos.

Se conseguíssemos ladrilhar usando triominós retos, então deveríamos usar $63/3 = 21$ triominós. E esses deveriam cobrir 21 quadrados cinza, 21 brancos e 21 pretos. Isso contradiz o fato de que esse tabuleiro contém 20 quadrados cinza, 21 pretos e 22 brancos. Portanto, não podemos cobrir esse tabuleiro usando triominós retos. ◀

O Papel dos Problemas Abertos

Muitos avanços em matemática têm sido feitos por pessoas que tentam resolver famosos problemas não resolvidos. Nos últimos 20 anos, muitos problemas foram finalmente resolvidos, como a demonstração da conjectura de teoria dos números feita há mais de 300 anos. Essa conjectura diz a verdade da sentença conhecida como O **Último Teorema de Fermat**.

TEOREMA 1 **ÚLTIMO TEOREMA DE FERMAT** A equação

$$x^n + y^n = z^n$$

não tem solução para x, y e z inteiros com $xyz \neq 0$ sempre que n for um inteiro com $n > 2$.

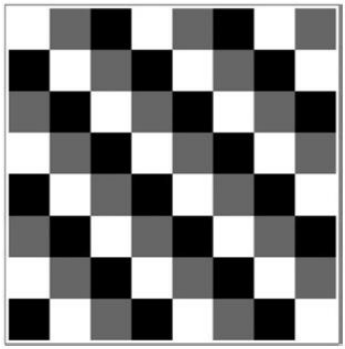


FIGURA 7 Colorindo os Quadrados de um Tabuleiro Standard com Três Cores.

Links ***Lembre-se:*** *A equação* $x^2 + y^2 = z^2$ tem infinitas soluções inteiras x, y e z; essas soluções são chamadas de triplas pitagóricas e correspondem aos comprimentos dos três lados de triângulos retos com comprimentos inteiros. Veja o Exercício 30.

Esse problema tem uma história fascinante. No século XVII, Fermat escreveu na margem da cópia dos trabalhos de Diofanto que ele tinha uma "maravilhosa demonstração" de que não existiam soluções inteiras de $x^n + y^n = z^n$ quando n é um inteiro maior que 2 com $xyz \neq 0$. No entanto, ele nunca publicou uma demonstração (Fermat não publicou quase nada), e nenhuma demonstração foi encontrada nos papéis que deixou quando morreu. Matemáticos procuraram uma demonstração por três séculos sem sucesso, embora muita gente estivesse convencida de que uma demonstração relativamente simples poderia ser encontrada. (Demonstrações para casos especiais foram encontradas, como a demonstração do caso $n = 3$ por Euler, e a demonstração do caso $n = 4$, do próprio Fermat.) Através dos anos, muitos matemáticos de renome pensaram que tinham demonstrado esse teorema. No século XIX, uma dessas tentativas falhas levou ao desenvolvimento de parte da teoria dos números, chamada de teoria algébrica dos números. Uma demonstração correta necessitou de centenas de páginas de matemática avançada e não tinha sido encontrada até a década de 1990, quando Andrew Wiles usou idéias desenvolvidas recentemente de uma área sofisticada da teoria dos números, chamada de teoria das curvas elípticas, para demonstrar o último teorema de Fermat. A pesquisa de Wiles para encontrar a demonstração do último teorema de Fermat, usando esta matemática poderosa, foi descrita na série *Nova* da televisão pública norte-americana e durou 10 anos! (O leitor interessado pode consultar [Ro05] para mais informações sobre o Último Teorema de Fermat e para referências adicionais sobre esse problema e sua resolução.)

Vamos agora mostrar um problema aberto simples de descrever, mas que parece difícil de resolver.

EXEMPLO 23 ***A Conjectura*** $3x+1$ Seja T a transformação que leva um número par x para x/2 e um número ímpar x para $3x + 1$. Uma conjectura famosa, conhecida como a **conjectura** $3x+1$, diz que para todos os números inteiros positivos x, quando aplicamos repetidamente a transformação T, vamos encontrar em algum ponto o inteiro 1. Por exemplo, começando com $x = 13$, encontramos $T(13) = 3 \cdot 13 + 1 = 40$, $T(40) = 40/2 = 20$, $T(20) = 20/2 = 10$, $T(10) = 10/2 = 5$, $T(5) = 3 \cdot 5 + 1 = 16$, $T(16) = 8$, $T(8) = 4$, $T(4) = 2$ e $T(2) = 1$. A conjectura $3x + 1$ foi verificada para todos os números inteiros até $5{,}6 \cdot 10^{13}$.

Links

A conjectura $3x + 1$ tem uma história interessante e tem atraído a atenção dos matemáticos desde a década de 1950. A conjectura apareceu muitas vezes e com muitos nomes diferentes, incluindo o problema Collatz, o algoritmo de Hasse, o problema de Ulam, o problema de Siracusa e o problema de Kakutani. Muitos matemáticos desviaram-se de seus trabalhos para gastar tempo dedicando-se inteiramente a essa conjectura. Isso levou à piada de que esse problema era parte de uma conspiração para diminuir o ritmo de desenvolvimento da pesquisa matemática norte-americana. Veja o artigo de Jeffrey Lagarias [La85] para uma discussão fascinante sobre esse problema e os resultados que já foram encontrados pelos matemáticos que se dedicaram a ele. ◀

No Capítulo 3, vamos descrever questões abertas adicionais sobre números primos. Estudantes já familiarizados com as noções básicas sobre primos podem querer explorar a Seção 3.4, onde estas questões abertas são discutidas. Vamos mencionar outras questões abertas importantes por todo o livro.

Métodos de Demonstrações Adicionais

Neste capítulo introduzimos os métodos básicos usados em demonstrações. Também descrevemos como usar esses métodos para demonstrar uma variedade de resultados. Vamos usar esses métodos de demonstração nos capítulos 2 e 3 para demonstrar resultados sobre conjuntos, funções, algoritmos e teoria dos números. Passando por esses teoremas, vamos demonstrar um famoso meta-teorema que diz que existe um problema que não pode ser resolvido usando qualquer procedimento. No entanto, existem muitos métodos de demonstração importantes ao lado desses que já cobrimos. Vamos introduzir alguns desses métodos mais adiante neste livro. Em particular, na Seção 4.1, vamos discutir indução matemática, que é um método extremamente usado para demonstrar sentenças da forma $\forall n P(n)$, em que o domínio consiste em todos os inteiros positivos. Na Seção 4.3, vamos introduzir indução estrutural, ou recursão, que pode ser usada para demonstrar resultados sobre conjuntos recursivamente definidos. Vamos usar o método de diagonalização de Cantor, que pode ser usado para demonstrar resultados sobre o tamanho de conjuntos infinitos, na Seção 2.4. No Capítulo 5, vamos introduzir a noção de demonstrações combinatórias, que podem ser usadas para demonstrar resultados usando argumentos de contagem. O leitor pode notar que no livro inteiro serão discutidas as atividades vistas nesta seção, incluindo muitos trabalhos excelentes de George Pólya ([Po61], [Po71], [Po90]).

Finalmente, note que não demos um procedimento que pode ser usado para demonstrar qualquer teorema em matemática. Este é um teorema profundo da lógica matemática que diz que não existe esse procedimento.

Exercícios

1. Demonstre que $n^2 + 1 \geq 2^n$ quando n é um número inteiro positivo com $1 \leq n \leq 4$.
2. Demonstre que não há cubos perfeitos positivos menores que 1.000 que são a soma dos cubos de dois números inteiros positivos.
3. Demonstre que se x e y são números reais, então $\max(x, y) + \min(x, y) = x + y$. [*Dica:* Use uma demonstração por casos, com os dois casos correspondentes a $x \geq y$ e $x < y$, respectivamente.]
4. Use uma demonstração por casos para mostrar que $\min(a, \min(b, c)) = \min(\min(a, b), c)$, sempre que a, b e c forem números reais.
5. Demonstre que a **desigualdade triangular**, que afirma que se x e y são números reais, então $|x| + |y| \geq |x + y|$ (em que $|x|$ representa o valor absoluto de x, que é igual a x *se* $x \geq 0$ e igual a $-x$ se $x < 0$).
6. Demonstre que há um número inteiro positivo que é igual à soma dos números inteiros positivos não excedentes a ele. Sua demonstração é construtiva ou não construtiva?
7. Demonstre que há 100 números inteiros positivos consecutivos que não são raízes quadradas perfeitas. Sua demonstração é construtiva ou não construtiva?
8. Demonstre que um desses números: $2 \cdot 10^{500} + 15$ ou $2 \cdot 10^{500} + 16$ não é um quadrado perfeito. Sua demonstração é construtiva ou não construtiva?
9. Demonstre que há um par de números inteiros consecutivos, tal que um desses números inteiros é um quadrado perfeito e o outro, um cubo perfeito.
10. Mostre que o produto de dois dos números $65^{1.000} - 8^{2.001} + 3^{177}$, $79^{1.212} - 9^{2.399} + 2^{2.001}$, e $24^{4.493} - 5^{8.192} + 7^{1.777}$ não é negativo. Sua demonstração é construtiva ou não construtiva? [*Dica:* Não tente avaliar esses números!]

11. Demonstre ou contrarie que há um número racional x e um número irracional y, tal que x^y é irracional.

12. Demonstre ou contrarie que se a e b são números racionais, então a^b é também racional.

13. Mostre que cada uma destas proposições pode ser usada para expressar o fato de que há um único elemento x, tal que $P(x)$ seja verdadeira. [Note que podemos escrever também esta proposição como $\exists! x\, P(x)$.]

a) $\exists x \forall y (P(y) \leftrightarrow x = y)$

b) $\exists x\, P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \rightarrow x = y)$

c) $\exists x (P(x) \wedge \forall y (P(y) \rightarrow x = y))$

14. Mostre que se a, b e c são números reais e $a \neq 0$, então há uma única solução para a equação $ax + b = c$.

15. Suponha que a e b sejam números inteiros ímpares, com $a \neq b$. Mostre que há um único número inteiro c, tal que $|a - c| = |b - c|$.

16. Mostre que se r é um número irracional, há um único número inteiro n, tal que a distância entre r e n é menor que 1/2.

17. Mostre que se n é um número inteiro ímpar, então há um único número inteiro k, tal que n é a soma de $k - 2$ e $k + 3$.

18. Demonstre que dado um número real x, existem números únicos n e ϵ, tal que $x = n + \epsilon$, n é um número inteiro e $0 \leq \epsilon < 1$.

19. Demonstre que dado um número real x, existem números únicos n e ϵ, tal que $x = n - \epsilon$, n é um número inteiro e $0 \leq \epsilon < 1$.

20. Use um raciocínio direto para mostrar que se x é um número real diferente de zero, então $x^2 + 1/x^2 \geq 2$. [*Dica:* Comece com a inequação $(x - 1/x)^2 \geq 0$ válida para todos os números reais diferentes de zero.]

21. A **média harmônica** de dois números reais x e y é igual a $2xy/(x + y)$. Computando as médias harmônica e geométrica dos diferentes pares dos números reais positivos, formule uma conjectura sobre seus tamanhos relativos e demonstre sua conjectura.

22. A **média quadrática** de dois números reais x e y é igual a $\sqrt{(x^2 + y^2)/2}$. Computando as médias aritmética e quadrática de pares diferentes de números reais positivos, formule uma conjectura sobre seus tamanhos relativos e demonstre sua conjectura.

***23.** Escreva os números 1, 2, . . . , $2n$ em uma lousa, onde n é um número inteiro ímpar. Escolha dois números quaisquer, j e k, escreva $|j - k|$ na lousa e apague j e k. Continue este processo até que um número inteiro seja escrito na lousa. Demonstre que este número inteiro deve ser ímpar.

***24.** Suponha que cinco números um e quatro zeros estão organizados em volta de um círculo. Entre dois números iguais, você insere um 0 e entre dois números diferentes você insere um 1 para produzir nove novos bits. Então, apague os nove bits originais. Mostre que, quando aplicar esse procedimento, nunca conseguirá nove zeros. [*Dica:* Trabalhe a partir do final, assumindo que você terminou com nove zeros.]

25. Formule uma conjectura sobre os dígitos decimais que aparecem como dígito final da quarta potência de um número inteiro. Demonstre sua conjectura, usando uma demonstração por casos.

26. Formule uma conjectura sobre os dígitos decimais finais do quadrado de um número inteiro. Demonstre sua conjectura, usando uma demonstração por casos.

27. Demonstre que não há um número inteiro positivo, tal que $n^2 + n^3 = 100$.

28. Comprove que não há soluções para os números inteiros x e y na equação $2x^2 + 5y^2 = 14$.

29. Comprove que não existem soluções para os números inteiros positivos x e y na equação $x^4 + y^4 = 625$.

30. Comprove que existem infinitas soluções para os números inteiros positivos x, y e z na equação $x^2 + y^2 = z^2$. [*Dica:* Considere $x = m^2 - n^2$, $y = 2mn$ e $z = m^2 + n^2$, em que m e n são números inteiros.]

31. Adapte a demonstração do Exemplo 4 da Seção 1.6 para demonstrar que se $n = abc$, em que a, b e c são números inteiros positivos, então $a \leq \sqrt[3]{n}$, $b \leq \sqrt[3]{n}$ ou $c \leq \sqrt[3]{n}$.

32. Comprove que $\sqrt[3]{2}$ é irracional.

33. Demonstre que entre dois números racionais há um número irracional.

34. Demonstre que entre um número racional e um irracional há um número irracional.

***35.** Considere $S = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$, em que $x_1, x_2, \ldots, x_n$ e $y_1, y_2, \ldots, y_n$ estão ordenados em duas seqüências de números reais positivos, cada uma com n elementos.

a) Mostre que S leva seu valor máximo em todas as ordenações das duas seqüências, quando as duas seqüências são aleatórias (ou seja, os elementos em cada seqüência estão em ordem não decrescente).

b) Mostre que S assume seu valor mínimo em todas as ordenações das suas seqüências, quando uma seqüência é sorteada entre as não decrescentes e a outra é sorteada entre as não crescentes.

36. Demonstre ou contrarie que se você tem um cântaro de água com capacidade para oito galões e dois cântaros vazios com capacidade para cinco galões e três galões, respectivamente, então você pode medir 4 galões pondo a água de um cântaro no outro.

37. Verifique a conjectura $3x + 1$ para os números inteiros abaixo.

a) 6 **b)** 7 **c)** 17 **d)** 21

38. Verifique a conjectura $3x + 1$ para os números inteiros abaixo.

a) 16 **b)** 11 **c)** 35 **d)** 113

39. Demonstre ou contrarie que você pode usar dominós para ladrilhar o tabuleiro de xadrez com os dois cantos adjacentes removidos (ou seja, cantos que não são opostos).

40. Comprove ou contrarie que você pode usar dominós para ladrilhar o tabuleiro de xadrez com os quatro cantos removidos.

41. Demonstre que você pode usar dominós para ladrilhar o tabuleiro de xadrez retangular com um número par de quadrados.

42. Comprove ou contrarie que você pode usar dominós para ladrilhar um tabuleiro de xadrez 5 por 5 com três cantos removidos.

43. Use uma demonstração por exaustão para mostrar que não existe um ladrilhamento que use dominós de um tabuleiro de xadrez 4 por 4 com lados opostos removidos. [*Dica:* Primeiro mostre que você pode assumir que os quadrados à esquerda superior e à direita inferior foram removidos. Numere os quadrados do tabuleiro original de 1 a 16, começando na primeira fila, da esquerda para a direita, então começando na esquerda da segunda fila e indo para a direita, e assim por diante. Remova os quadrados 1 e 16. Para começar a demonstração, note que o quadrado 2 está coberto pelo dominó na horizontal, que cobre os quadrados 2 e 3, ou verticalmente, que cobre os quadrados 2 e 6. Considere cada um desses casos separadamente e trabalhe com todos os subcasos que aparecerem.]

***44.** Demonstre que quando um quadrado branco e um quadrado preto são removidos de um tabuleiro de xadrez 8 por 8 (colorido como no texto), você pode ladrilhar o restante dos quadrados do tabuleiro usando dominós. [*Dica:* Mostre que, quando um quadrado preto e um branco são removidos, cada parte da divisão das células restantes formada inserindo as barrinhas, como mostra a figura, pode ser coberta pelos dominós.]

45. Mostre que, removendo dois quadrados brancos e dois pretos de um tabuleiro de xadrez 8 por 8 (colorido como no texto), torna-se impossível ladrilhar os quadrados restantes com dominós.

***46.** Encontre todos os quadrados, se eles existirem, em um tabuleiro de xadrez 8 por 8, para que o tabuleiro obtido pela remoção desses quadrados possa ser ladrilhado utilizando-se triominós. [*Dica:* Primeiro use argumentos que se baseiam na cor e na rotação das peças eliminadas, tanto quanto for possível.]

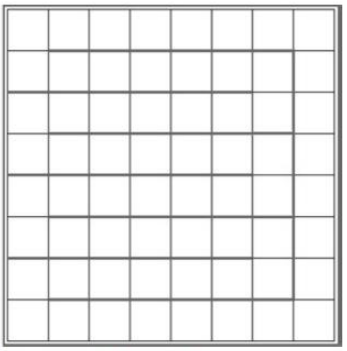


Figura para o Exercício 44.

***47.** **a)** Desenhe cada um dos cinco diferentes tetraominós, em que um tetraominó é um poliominó com quatro quadrados.

b) Para cada um dos cinco tetraominós, demonstre ou contrarie que é possível ladrilhar um tabuleiro de xadrez usando essas peças.

***48.** Demonstre ou contrarie que é possível ladrilhar um tabuleiro de xadrez 10 por 10 usando tetraominós retos.

Termos-chave e Resultados

TERMOS

proposição: uma sentença que é verdadeira ou falsa

variável proposicional: variável que representa uma proposição

valor-verdade: verdadeiro ou falso

$\neg p$ (negação de p): proposição com o valor-verdade oposto do valor-verdade de p

operadores lógicos: operadores usados para combinar proposições

proposições compostas: proposição construída pela combinação de proposições usando operadores lógicos

tabela-verdade: tabela que mostra os valores-verdade das proposições

$p \vee q$ (disjunção de p e q): proposição "p ou q", que é verdadeira se e somente se ao menos uma entre p e q for verdadeira

$p \wedge q$ (conjunção de p e q): proposição "p e q", que é verdadeira se e somente se ambas p e q forem verdadeiras

$p \oplus q$ (ou exclusivo de p e q): proposição "p XOR q", que é verdadeira se e somente se exatamente uma entre p e q for verdadeira

$p \rightarrow q$ (p implica q): proposição "se p, então q", que é falsa se e somente se p for verdadeira e q for falsa

oposta de $p \rightarrow q$: sentença condicional $q \rightarrow p$

contrapositiva de $p \rightarrow q$: sentença condicional $\neg q \rightarrow \neg p$

inversa de $p \rightarrow q$: sentença condicional $\neg p \rightarrow \neg q$

$p \leftrightarrow q$ (bicondicional): proposição "p se e somente se q", que é verdadeira se e somente se p e q tiverem o mesmo valor-verdade

bit: ou um 0 ou um 1

variável booleana: variável que tem um valor 0 ou 1

operação binária (operação bit): operação em um bit ou em bits

seqüência binária: lista de bits

operações binárias: operações entre seqüências de bits, operando cada um de seus bits correspondentes

tautologia: proposição composta que é sempre verdadeira

contradição: proposição composta que é sempre falsa

contingência: proposição composta que é às vezes verdadeira, às vezes falsa

proposições compostas consistentes: proposições compostas para as quais existe uma valoração para as variáveis que faz todas as proposições verdadeiras

proposições compostas logicamente equivalentes: proposições compostas que sempre têm o mesmo valor-verdade

predicado: parte de uma sentença que atribui uma propriedade ao sujeito

função proposicional: sentença com uma ou mais variáveis que se torna uma proposição quando cada uma das variáveis recebe um valor ou é fechada por um quantificador

domínio (ou universo) de discurso: valores que uma variável em uma função proposicional pode ter

$\exists x\, P(x)$ (quantificação existencial de $P(x)$): proposição que é verdadeira se e somente se existir um x no domínio, tal que $P(x)$ é verdadeira

$\forall x\, P(x)$ (quantificação universal de $P(x)$): proposição que é verdadeira se e somente se $P(x)$ for verdadeira para todo x no domínio

expressões logicamente equivalentes: expressões que têm o mesmo valor-verdade, não importando qual função proposicional e qual domínio são usados

variável livre: variável não ligada em uma função proposicional

variável ligada: variável que é quantificada

escopo de um quantificador: porção de uma sentença em que o quantificador liga suas variáveis

argumento: seqüência de sentenças

forma de argumento: seqüência de proposições compostas que envolve variáveis proposicionais

premissa: sentença, em um argumento, ou forma de argumento, que não seja a final

conclusão: sentença final de um argumento ou forma de argumento

forma válida de argumento: seqüência de proposições compostas que envolve variáveis proposicionais, tal que a verdade de todas as premissas implica a verdade da conclusão

argumento válido: argumento com uma forma válida de argumento

regra de inferência: forma válida de argumento que pode ser usada na apresentação de argumentos válidos

falácia: forma inválida de argumento freqüentemente usado incorretamente como uma regra de inferência (ou, às vezes, mais geralmente, um argumento incorreto)

raciocínio circular ou que carrega a pergunta: raciocínio em que um ou mais passos se baseiam na verdade da sentença que está sendo demonstrada

teorema: afirmação matemática que pode ser demonstrada verdadeira

conjectura: afirmação matemática proposta como verdade, mas que ainda não foi provada

demonstração: modelo de um teorema que é verdadeiro

axioma: sentença que é assumida como verdadeira e que pode ser usada como base para demonstrar teoremas

lema: teorema usado para demonstrar outros teoremas

corolário: proposição que pode ser demonstrada como uma conseqüência de um teorema que está sendo demonstrado

demonstração por vacuidade: demonstração de que $p \rightarrow q$ é verdadeira com base no fato de que p é falsa

demonstração por trivialização: demonstração de que $p \rightarrow q$ é verdadeira com base no fato de que q é verdadeira

demonstração direta: demonstração de que $p \rightarrow q$ é verdadeira, que procede mostrando que q deve ser verdadeira quando p é verdadeira

demonstração por contraposição: demonstração de que $p \rightarrow q$ é verdadeira, que procede mostrando que p deve ser falsa quando q é falsa

demonstração por contradição: demonstração de que p é verdadeira com base na verdade do condicional $\neg p \rightarrow q$, na qual q é uma contradição

demonstração por exaustão: demonstração que estabelece um resultado, verificando uma lista de todos os casos

demonstração por casos: demonstração quebrada em casos separados em que esses casos cobrem todas as possibilidades

sem perda de generalidade: quando se assume em uma demonstração que é possível demonstrar um teorema, reduzindo o número de casos necessários para a demonstração

contra-exemplo: elemento x, tal que $P(x)$ é falsa

demonstração de existência construtiva: demonstração de que um elemento com uma propriedade específica existe que explicitamente encontra um tal elemento

demonstração de existência não construtiva: demonstração de que um elemento com uma propriedade específica existe que não encontra explicitamente um tal elemento

número racional: número que pode ser expresso como a razão de dois inteiros p e q, tal que $q \neq 0$

demonstração de unicidade: demonstração de que existe exatamente um elemento que satisfaz uma propriedade específica

RESULTADOS

As equivalências lógicas dadas nas tabelas 6, 7 e 8 da Seção 1.2.

Leis de De Morgan para quantificadores.

Regras de inferência para os cálculos proposicionais.

Regras de inferência para sentenças quantificadas.

Questões de Revisão

1. **a)** Defina a negação de uma proposição.
   b) Qual é a negação de "Este é um curso entediante"?
2. **a)** Defina (usando tabelas-verdade) a disjunção, conjunção, ou exclusivo, condicional e bicondicional das proposições p e q.
   b) Quais são: disjunção, conjunção, ou exclusivo, condicional e bicondicional das proposições "Eu vou ao cinema esta noite" e "Eu vou terminar minha lição de casa de matemática discreta"?
3. **a)** Descreva pelo menos cinco modos diferentes de escrever o condicional $p \rightarrow q$ em português.
   b) Defina a oposta e a contrapositiva de uma sentença condicional.
   c) Dê a oposta e a contrapositiva da sentença condicional "Se amanhã fizer sol, então eu vou fazer uma trilha na mata".
4. **a)** O que significa duas proposições serem logicamente equivalentes?

b) Descreva as maneiras diferentes de mostrar que duas proposições compostas são logicamente equivalentes.

c) Mostre pelo menos dois modos diferentes em que as proposições compostas $\neg p \vee (r \rightarrow \neg q)$ e $\neg p \vee \neg q \vee \neg r$ são equivalentes.

5. *(Depende do conjunto de exercícios da Seção 1.2)*

a) Dada uma tabela-verdade, explique como usar a forma normal disjuntiva para construir proposições compostas com esta tabela-verdade.

b) Explique por que a parte (a) mostra que os operadores $\wedge$, $\vee$ e $\neg$ são funcionalmente completos.

c) Existe um operador, tal que o conjunto com apenas esse operador seja funcionalmente completo?

6. O que são quantificações universal e existencial de um predicado $P(x)$? Quais são suas negações?

7. **a)** Qual a diferença entre a quantificação $\exists x \forall y\, P(x, y)$ e $\forall y \exists x\, P(x, y)$, em que $P(x, y)$ é um predicado?

b) Dê um exemplo de um predicado $P(x, y)$, tal que $\exists x \forall y\, P(x, y)$ e $\forall y \exists x\, P(x, y)$ têm diferentes valores-verdade.

8. Descreva o que chamamos de argumento válido em lógica proposicional e mostre que o argumento "Se a Terra é plana, então podemos navegar até a borda da Terra", "Você não pode navegar até a borda da Terra", portanto "A Terra não é plana" é um argumento válido.

9. Use as regras de inferência para mostrar que as premissas "Todas as zebras têm listras" e "Mark é uma zebra" são verdadeiras, então a conclusão "Mark tem listras" é verdadeira.

10. **a)** Descreva o que se entende por demonstração direta, demonstração por contraposição e demonstração por contradição de uma sentença condicional $p \rightarrow q$.

b) Dê uma demonstração direta, uma demonstração por contraposição e uma demonstração por contradição da sentença: "Se n é par, então $n + 4$ é par".

11. **a)** Descreva um modo de provar a bicondicional $p \leftrightarrow q$.

b) Demonstre a sentença: "O inteiro $3n + 2$ é ímpar se e somente se o inteiro $9n + 5$ é par, em que n é um inteiro".

12. Para demonstrar que as sentenças p_1, p_2, p_3 e p_4 são equivalentes, é suficiente mostrar que as sentenças condicionais $p_4 \rightarrow p_2$, $p_3 \rightarrow p_1$ e $p_1 \rightarrow p_2$ são válidas? Se não, mostre outro conjunto de condicionais que podem ser usados para mostrar que as 4 sentenças são equivalentes.

13. **a)** Suponha que uma sentença da forma $\forall x P(x)$ é falsa. Como isso pode ser demonstrado?

b) Mostre que a sentença "Para todo inteiro positivo n, $n^2 \geq 2n$" é falsa.

14. Qual a diferença entre uma demonstração de existência construtiva e uma não construtiva? Dê um exemplo de cada.

15. Quais são os elementos de uma demonstração para a existência de um único elemento x, tal que $P(x)$, em que $P(x)$ é uma função proposicional?

16. Explique como uma demonstração por casos pode ser usada para demonstrar um resultado sobre valores absolutos, tal como o fato de que $|xy| = |x||y|$ para todos os reais x e y.

Exercícios Complementares

1. Seja p a proposição "Eu vou fazer todo exercício deste livro" e q a proposição "Eu vou tirar 'A' neste curso". Expresse cada uma delas como uma combinação de p e q.

a) Eu vou tirar "A" neste curso somente se eu fizer todos os exercícios deste livro.

b) Eu vou tirar "A" neste curso e vou fazer todos os exercícios deste livro.

c) Ou eu não vou tirar "A" neste curso ou eu não vou fazer todos os exercícios deste livro.

d) Para eu tirar "A" neste curso é necessário e suficiente que eu faça todos os exercícios deste livro.

2. Faça a tabela-verdade da proposição composta $(p \vee q) \rightarrow (p \wedge \neg r)$.

3. Mostre que estas proposições compostas são tautologias.

a) $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$

b) $((p \vee q) \wedge \neg p) \rightarrow q$

4. Dê a oposta, a contrapositiva e a inversa dessas sentenças condicionais.

a) Se está chovendo, então eu vou de carro para o trabalho.

b) Se $|x| = x$, então $x \geq 0$.

c) Se n é maior que 3, então n^2 é maior que 9.

5. Dada uma sentença condicional $p \rightarrow q$, encontre a oposta de sua inversa, a oposta de sua oposta e a oposta de sua contrapositiva.

6. Dada uma sentença condicional $p \rightarrow q$, encontre a inversa de sua inversa, a inversa de sua oposta e a inversa de sua contrapositiva.

7. Encontre uma proposição composta que envolva as variáveis proposicionais p, q, r e s que é verdadeira quando exatamente três dessas variáveis proposicionais são verdadeiras e é falsa, caso contrário.

8. Mostre que estas sentenças são inconsistentes: "Se Sergei aceitar o trabalho oferecido, então ele terá mais tempo em casa", "Se Sergei aceitar o trabalho oferecido, então ele receberá um salário maior", "Se Sergei terá mais tempo em casa, então ele não receberá um salário maior".

9. Mostre que estas sentenças são inconsistentes: "Se Miranda não tiver o curso de matemática discreta, então ela não se formará", "Se Miranda não se formar, então ela não obterá um bom trabalho", "Se Miranda ler este livro, então ela obterá um bom trabalho", "Miranda não tem o curso de matemática discreta, mas leu este livro".

10. Suponha que você encontre três pessoas A, B e C, na ilha dos cavaleiros e dos bandidos descrita no Exemplo 18 da Seção 1.1. O que são A, B e C se A diz "Eu sou um bandido e B é um cavaleiro" e B diz "Exatamente um de nós três é um cavaleiro"?

11. (Adaptado de [Sm78]) Suponha que em uma ilha existem três tipos de pessoas: cavaleiros, bandidos e normais. Cavaleiros sempre dizem a verdade, bandidos sempre mentem e normais às vezes mentem, às vezes dizem a verdade. Detetives que investigam um crime questionaram três habitantes — Ana, Bela e Clarice. Os detetives sabiam que uma das três teria cometido o crime, mas não sabiam qual. Eles também sabiam que a criminosa era uma cavaleira e as outras duas não. Adicionalmente, os detetives gravaram estas sentenças: Ana: "Eu sou inocente", Bela: "O que Ana diz é verdade", Clarice: "Bela não é normal". Depois de analisar essas informações, os detetives identificaram positivamente o culpado. Quem era?

12. Mostre que se S é uma proposição, em que S é a sentença condicional "Se S é verdadeira, então unicórnios existem", então "Unicórnios existem" é verdadeira. Mostre que disso segue que S não pode ser uma proposição. (Esse paradoxo é conhecido como *paradoxo de Löb*.)

13. Mostre que o argumento com premissas "O saci-pererê é uma personagem real", "O saci-pererê não é uma personagem real" e a conclusão "Você pode encontrar ouro no final do arco-íris" é um argumento válido. Isso mostra que a conclusão é verdadeira?

14. Seja $P(x)$ a sentença "O estudante x sabe cálculo" e seja $Q(y)$ a sentença "A classe y contém um estudante que sabe cálculo". Expresse cada uma das quantificações de $P(x)$ e $Q(y)$.

a) Algum estudante sabe cálculo.

b) Nenhum estudante sabe cálculo.

c) Toda classe tem um estudante que sabe cálculo.

d) Todo estudante em toda classe sabe cálculo.

e) Existe ao menos uma classe sem algum estudante que sabe cálculo.

15. Seja $P(m, n)$ a sentença "m divide n", em que o domínio para as variáveis consiste em todos os inteiros positivos. (Por "m divide n", entendemos que $n = km$ para algum inteiro k.) Determine o valor-verdade de cada uma destas sentenças.

a) $P(4, 5)$ **b)** $P(2, 4)$

c) $\forall m \, \forall n P(m, n)$ **d)** $\exists m \, \forall n P(m, n)$

e) $\exists n \, \forall m P(m, n)$ **f)** $\forall n \, P(1, n)$

16. Encontre um domínio para os quantificadores em $\exists x \exists y (x \neq y \wedge \forall z \, ((z \neq x) \wedge (z \neq y))$, tal que essa sentença seja verdadeira.

17. Encontre um domínio para os quantificadores em $\exists x \exists y (x \neq y \wedge \forall z \, ((z = x) \vee (z = y)))$, tal que essa sentença seja falsa.

18. Use quantificadores existencial e universal para expressar a sentença "Ninguém tem mais que três avós" usando a função proposicional $G(x, y)$, que representa "x é a avó de y".

19. Use quantificadores existencial e universal para expressar a sentença "Todos têm exatamente dois pais biológicos" usando a função proposicional $P(x, y)$, que representa "x é pai (ou mãe) biológico de y".

20. O quantificador $\exists n$ denota "existem exatamente n", portanto $\exists_n x \, P(x)$ significa que existem exatamente n elementos do domínio, tal que $P(x)$ é verdadeira. Determine o valor-verdade destas sentenças, em que o domínio consiste em todos os números reais.

a) $\exists_0 x(x^2 = -1)$ **b)** $\exists_1 x(|x| = 0)$

c) $\exists_2 x(x^2 = 2)$ **d)** $\exists_3 x(x = |x|)$

21. Expresse cada uma destas sentenças usando quantificadores universal e existencial e lógica proposicional, em que $\exists_n$ é definido no Exercício 20.

a) $\exists_0 x \, P(x)$

b) $\exists_1 x \, P(x)$

c) $\exists_2 x \, P(x)$

d) $\exists_3 x \, P(x)$

22. Seja $P(x, y)$ uma função proposicional. Mostre que $\exists x \, \forall y \, P(x, y) \rightarrow \forall y \, \exists x \, P(x, y)$ é uma tautologia.

23. Sejam $P(x)$ e $Q(x)$ funções proposicionais. Mostre que $\exists x \, (P(x) \rightarrow Q(x))$ e $\forall x \, P(x) \rightarrow \exists x \, Q(x)$ sempre têm o mesmo valor-verdade.

24. Se $\forall y \, \exists x \, P(x, y)$ é verdadeira, segue que, necessariamente, $\exists x \, \forall y \, P(x, y)$ é verdadeira?

25. Se $\forall x \, \exists y \, P(x, y)$ é verdadeira, segue que, necessariamente, $\exists x \, \forall y \, P(x, y)$ é verdadeira?

26. Encontre as negações destas sentenças.

a) Se nevar hoje, então eu vou esquiar amanhã.

b) Toda pessoa nesta classe entende indução matemática.

c) Algum estudante desta classe não gosta de matemática discreta.

d) Em toda classe de matemática existe algum estudante que está com sono durante as aulas.

27. Expresse esta sentença usando quantificadores: "Todo estudante nesta classe teve algum curso de ciências matemáticas em todo o departamento da escola".

28. Expresse esta sentença usando quantificadores: "Existe um prédio no campus de alguma faculdade brasileira, tal que todas as suas salas são pintadas de branco".

29. Expresse a sentença: "Existe exatamente um estudante nesta classe que teve exatamente um curso de matemática nesta faculdade" usando quantificadores de unicidade. Depois expresse essa sentença usando quantificadores, sem o uso dos quantificadores de unicidade.

30. Descreva uma regra de inferência que pode ser usada para demonstrar que existem exatamente dois elementos x e y em um domínio, tal que $P(x)$ e $P(y)$ são verdadeiras. Expresse essa regra de inferência em português.

31. Use regras de inferência para mostrar que se as premissas $\forall x \, (P(x) \rightarrow Q(x))$, $\forall x \, (Q(x) \rightarrow R(x))$ e $\neg R(a)$, em que a está no domínio, são verdadeiras, então a conclusão $\neg P(a)$ é verdadeira.

32. Demonstre que se x^3 é irracional, então x é irracional.
33. Demonstre que se x é irracional e $x \geq 0$, então $\sqrt{x}$ é irracional.
34. Demonstre que dado um inteiro não negativo n, existe um único não negativo m, tal que $m^2 \leq n < (m+1)^2$.
35. Demonstre que existe um inteiro m, tal que $m^2 > 10^{1.000}$. Sua demonstração é construtiva ou não construtiva?
36. Demonstre que existe um inteiro positivo que pode ser escrito como a soma dos quadrados de inteiros positivos de duas maneiras diferentes. (Use computador ou calculadora para acelerar seu trabalho.)
37. Negue a sentença: todo inteiro positivo é a soma dos cubos de oito inteiros não negativos.
38. Negue a sentença: todo inteiro positivo é a soma de, no máximo, dois quadrados e um cubo de inteiros não negativos.
39. Negue a sentença: todo inteiro positivo é a soma de 36 quintas potências de inteiros não negativos.
40. Assumindo a verdade do teorema que diz que $\sqrt{n}$ é irracional toda vez que n é um inteiro positivo que não é um quadrado perfeito, demonstre que $\sqrt{2} + \sqrt{3}$ é irracional.

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dados os valores-verdade das proposições p e q, encontre os valores-verdade de conjunção, disjunção, ou exclusivo, sentença condicional e bicondicional dessas proposições.
2. Dadas duas seqüências de bits de comprimento n, encontre as seqüências resultantes das operações *AND*, *OR* e *XOR* dessas seqüências.
3. Dados os valores-verdade das proposições p e q em lógica Fuzzy, encontre os valores-verdade da disjunção e da conjunção de p e q (veja os exercícios 40 e 41 da Seção 1.1).
***4.** Dados os inteiros positivos m e n, jogue o jogo Chomp interativamente.
***5.** Dado um pedaço de um tabuleiro, procure modos de ladrilhar esse tabuleiro com vários tipos de poliominós, incluindo dominós, os dois tipos de triominós e poliominós maiores.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Procure inteiros positivos que não são a soma dos cubos de nove inteiros positivos diferentes.
2. Procure inteiros positivos maiores que 79 que não são a soma de quartas potências de 18 inteiros positivos.
3. Encontre quantos inteiros positivos você puder, tal que eles possam ser escritos como a soma de cubos de inteiros positivos, de duas maneiras diferentes, começando com o número 1.729.
***4.** Tente encontrar estratégias vencedoras para o jogo Chomp para diferentes configurações iniciais de biscoitos.
***5.** Procure maneiras de ladrilhar tabuleiros com poliominós.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Discuta paradoxos lógicos, incluindo o paradoxo de Epimenides, o Cretense, o paradoxo da carta de Jourdain e o paradoxo do barbeiro e mostre como eles podem ser resolvidos.
2. Descreva como a lógica fuzzy está sendo aplicada na prática. Consulte um ou mais dos recentes livros de lógica fuzzy escritos para o público em geral.
3. Descreva as regras básicas do *WFF'N PROOF*, *The Game of Modern Logic*, desenvolvido por Layman Allen. Dê exemplos de alguns dos jogos incluídos no *WFF'N PROOF*.
4. Leia alguns dos escritos de Lewis Carroll sobre lógica simbólica. Descreva em detalhes alguns dos modelos usados para representar argumentos lógicos e as regras de inferência que ele usou nesses argumentos.
5. Extenda a discussão do Prolog dada na Seção 1.3, explicando mais profundamente como o Prolog faz suas resoluções.
6. Discuta algumas técnicas usadas em lógica computacional, incluindo a regra de Skolem.
7. "Demonstração de teoremas automatizada" é uma tentativa de usar computadores para demonstrar mecanicamente

teoremas. Discuta os objetivos e as aplicações da demonstração automatizada de teoremas e o progresso feito no desenvolvimento de provadores automáticos.

8. Descreva como computação de DNA tem sido usada para resolver instâncias do problema de satisfação.

9. Discuta o que é conhecido sobre estratégias vencedoras do jogo Chomp.

10. Descreva vários aspectos de estratégia de demonstrações discutidos por George Pólya em seus escritos sobre raciocínio, incluindo [Po62], [Po71] e [Po90].

11. Descreva alguns problemas e resultados sobre maneiras de ladrilhar como poliominós, como os descritos em [Go94] e [Ma91], por exemplo.

CAPÍTULO

2 Estruturas Básicas: Conjuntos, Funções, Seqüências e Somatórios

2.1 Conjuntos
2.2 Operações com Conjuntos
2.3 Funções
2.4 Seqüências e Somatórios

Grande parte da matemática discreta é voltada para o estudo das estruturas discretas, usadas para representar objetos discretos. Muitas estruturas discretas importantes foram construídas usando conjuntos, que são coleções de objetos. Entre as estruturas discretas construídas com conjuntos temos a combinatória, coleção não ordenada de objetos extremamente utilizada em contagem; relações, conjuntos de pares ordenados que representam as relações entre objetos; gráficos, conjuntos de vértices e arcos que conectam esses vértices; e as máquinas de estado finito, usadas como modelo de máquinas computacionais. Esses são alguns dos tópicos que estudaremos em capítulos posteriores.

O conceito de função é extremamente importante em matemática discreta. Uma função determina, para cada elemento de um conjunto, exatamente um elemento de um conjunto. Funções desempenham importantes papéis na matemática discreta. Elas são usadas para representar a complexidade computacional de algoritmos, para estudar o tamanho dos conjuntos, para contar objetos e em muitos outros casos. Estruturas úteis como as seqüências e as cadeias são tipos especiais de funções. Neste capítulo, vamos introduzir a noção de seqüência, que representa uma lista ordenada de elementos. Introduziremos alguns tipos importantes de seqüências e direcionaremos o problema de identificação de um modelo para o termo geral de uma seqüência a partir de seu primeiro termo. Usando a noção de seqüência, definiremos o que se entende por um conjunto contável, ou seja, quando podemos listar todos os elementos do conjunto em uma seqüência.

Em nosso estudo sobre matemática discreta, geralmente vamos adicionar termos consecutivos de uma seqüência de nú139meros. Como somar termos de uma seqüência, assim como inserir outros conjuntos de números, é uma ocorrência comum; uma notação especial foi desenvolvida para a adição de tais termos. Nesta seção, introduziremos a notação usada para expressar somatórios. Desenvolveremos fórmulas para certos tipos de somatórios. Esses somatórios aparecem no estudo de matemática discreta como, por exemplo, quando analisamos o número de passos que um procedimento usa para classificar uma lista de números em ordem crescente.

2.1 Conjuntos

Introdução

Nesta seção, estudaremos a estrutura discreta fundamental sob a qual todas as outras estruturas discretas são construídas: o conjunto. Os conjuntos são usados para agrupar objetos. Geralmente, os objetos de um conjunto têm propriedades semelhantes. Por exemplo, todos os estudantes que são inscritos em uma faculdade formam um conjunto. Da mesma forma, todos os estudantes que assistem ao curso de matemática discreta em qualquer faculdade formam um conjunto. Além disso, aqueles estudantes inscritos em sua faculdade que estão cursando matemática discreta formam um conjunto que pode ser obtido a partir dos elementos comuns dos dois primeiros conjuntos. A linguagem dos conjuntos é uma maneira de estudar tais coleções em sua organização. Forneceremos agora uma definição de conjunto. Essa definição é uma definição intuitiva, que não faz parte da teoria formal dos conjuntos.

DEFINIÇÃO 1 Um *conjunto* é uma coleção não ordenada de objetos.

Links

Note que o termo *objeto* foi usado sem especificar o que é um objeto. Essa descrição de um conjunto como uma coleção de objetos, baseada em uma noção intuitiva, foi expressa pela primeira vez pelo matemático alemão Georg Cantor em 1895. A teoria que resulta dessa definição intuitiva

Auto-avaliação

de conjunto e o uso da noção intuitiva que para qualquer propriedade sempre há um conjunto que contém os objetos com essa propriedade, leva a um **paradoxo**, ou inconsistência lógica. Isto foi mostrado pelo filósofo inglês Bertrand Russell em 1902 (veja o Exercício 38 para uma descrição de um desses paradoxos). Essas inconsistências lógicas podem ser evitadas pela construção da teoria dos conjuntos, começando com axiomas. Usaremos a versão original de Cantor da teoria dos conjuntos, conhecida como **teoria ingênua dos conjuntos**, sem desenvolver uma versão axiomática da teoria dos conjuntos, porque todos os conjuntos considerados neste livro podem ser tratados consistentemente usando a teoria original de Cantor.

DEFINIÇÃO 2 Os objetos no conjunto são chamados de *elementos,* ou *membros,* do conjunto. Diz-se que os elementos *pertencem* ao conjunto.

Introduziremos agora a notação usada para descrever pertinência em conjuntos. Escrevemos $a \in A$ para indicar que a é um elemento do conjunto A. A notação $a \notin A$ indica que a não é um elemento do conjunto A. Note que as letras minúsculas são geralmente usadas para indicar elementos dos conjuntos e as maiúsculas para indicar os conjuntos.

Há muitas maneiras de descrever um conjunto. Uma maneira é listar todos os seus elementos, quando possível. A notação usada para indicar todos os membros do conjunto listados é por meio de chaves. Por exemplo, a notação $\{a, b, c, d\}$ representa o conjunto com seus quatro elementos a, b, c e d.

EXEMPLO 1 O conjunto V de todas as vogais do alfabeto da língua inglesa pode ser escrito como $V = \{a, e, i, o, u\}$. ◀

EXEMPLO 2 O conjunto O dos números inteiros positivos ímpares menores que 10 pode ser expresso por $O = \{1, 3, 5, 7, 9\}$. ◀

EXEMPLO 3 Embora os conjuntos sejam normalmente usados para agrupar elementos com propriedades semelhantes, não há nada que impeça um conjunto de ter elementos que não contenham relação alguma. Por exemplo, {*a*, 2, Carlos, São Paulo} é o conjunto que contém os quatro elementos *a*, 2, Carlos e São Paulo. ◀

Algumas vezes as chaves são usadas para descrever um conjunto sem listar todos os seus elementos. Alguns elementos do conjunto são listados e, então, os *três pontos* (. . .) são usados quando o modelo geral dos elementos é óbvio.

EXEMPLO 4 O conjunto de números inteiros positivos menores que 100 pode ser indicado por $\{1, 2, 3, \ldots, 99\}$. ◀

Exemplos Extras

Outra maneira de descrever um conjunto é usar a notação de **construção do conjunto**. Caracterizamos todos aqueles elementos do conjunto estabelecendo a propriedade, ou propriedades, que eles devem ter. Por exemplo, o conjunto O de todos os números inteiros positivos e ímpares menores que 10 pode ser escrito como

$$O = \{x \mid x \text{ é um número inteiro positivo e ímpar menor que } 10\},$$

ou, especificando o universo do conjunto, como

$$O = \{ x \in \mathbf{Z}^+ \mid x \text{ é impar e } x < 10\}.$$

Usamos geralmente esse tipo de notação para descrever conjuntos quando é impossível listar todos os seus elementos. Por exemplo, o conjunto $\mathbf{Q}^+$ de todos os números racionais positivos pode ser escrito como

$$\mathbf{Q}^+ = \{ x \in \mathbf{R} \mid x = p/q\}, \text{ para todos inteiros positivos } p \text{ e } q.$$

Os próximos conjuntos, cada um usando uma letra em negrito, têm um importante papel em matemática discreta:

$\mathbf{N} = \{0, 1, 2, 3, \ldots\}$, o conjunto dos **números naturais**
$\mathbf{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, o conjunto dos **números inteiros**
$\mathbf{Z}^+ = \{1, 2, 3, \ldots\}$, o conjunto dos **números inteiros positivos**
$\mathbf{Q} = \{p/q \mid p \in \mathbf{Z}, q \in \mathbf{Z}, \text{e } q \neq 0\}$, o conjunto dos **números racionais**
$\mathbf{R}$, o conjunto dos **números reais**

(Note que algumas pessoas não consideram 0 como um número natural, então tenha o cuidado de checar como o termo *números naturais* é usado ao ler outros livros.)

Os conjuntos podem ter outros conjuntos como elementos, conforme o Exemplo 5 ilustra.

EXEMPLO 5 O conjunto $\{\mathbf{N}, \mathbf{Z}, \mathbf{Q}, \mathbf{R}\}$ é um conjunto que contém quatro elementos, sendo cada um deles um conjunto. Os quatro elementos desse conjunto são $\mathbf{N}$, o conjunto dos números naturais; $\mathbf{Z}$, o conjunto dos números inteiros; $\mathbf{Q}$, o conjunto dos números racionais; e $\mathbf{R}$, o conjunto dos números reais. ◀

Lembre-se: Note que o conceito de tipo de dado, ou tipo, em ciência da computação é construído com o conceito de conjunto. Em particular, um **tipo de dado** ou **tipo** é o nome de um conjunto, acrescido do conjunto de operações que podem ser formadas com os objetos desse conjunto. Por exemplo, *booleano* é o nome do conjunto $\{0, 1\}$ acrescido dos operadores para um ou mais elementos desse conjunto, como E, OU e NÃO.

Como muitas proposições matemáticas afirmam que duas coleções especificamente diferentes são o mesmo conjunto, precisamos entender o que significa dois conjuntos serem iguais.

DEFINIÇÃO 3 Dois conjuntos são *iguais* se e somente se eles têm os mesmos elementos. Ou seja, se A e B são conjuntos, então A e B são iguais se e somente se $\forall x (x \in A \leftrightarrow x \in B)$. Escrevemos $A = B$ se A e B forem conjuntos iguais.

EXEMPLO 6 Os conjuntos $\{1, 3, 5\}$ e $\{3, 5, 1\}$ são iguais, porque eles têm os mesmos elementos. Note que a ordem na qual os elementos de um conjunto são listados não é relevante. Note também que não é relevante se um elemento de um conjunto é listado mais de uma vez, então $\{1, 3, 3, 3, 5, 5, 5, 5\}$ é o mesmo que o conjunto $\{1, 3, 5\}$ porque eles apresentam os mesmos elementos. ◀

Os conjuntos podem ser representados graficamente usando diagramas de Venn, nomeado em homenagem ao matemático inglês John Venn, que introduziu seu uso em 1881. Nos diagramas de Venn, o **conjunto universo** U, que contém todos os objetos em consideração, é representado por um retângulo. (Note que o conjunto universo varia dependendo dos objetos de interesse.) Dentro desse retângulo, círculos ou outras figuras geométricas são usadas para representar os conjuntos. Algumas vezes, pontos são usados para representar elementos particulares do conjunto. Os diagramas de Venn são geralmente usados para indicar as relações entre conjuntos. Mostraremos como um diagrama de Venn pode ser usado no Exemplo 7.

Links

GEORG CANTOR (1845–1918) Georg Cantor nasceu em St. Petersburgo, na Rússia, onde seu pai foi um comerciante bem-sucedido. Cantor desenvolveu seu interesse em matemática na juventude. Iniciou seus estudos universitários em Zurique, em 1862, mas quando seu pai morreu, ele deixou a cidade. Continuou seus estudos universitários na Universidade de Berlim, em 1863, onde estudou com os eminentes matemáticos Weierstrass, Kummer e Kronecker. Ele recebeu seu título de doutor em 1867, depois de escrever uma dissertação sobre a teoria dos números. Cantor assumiu um cargo na Universidade de Halle em 1869, onde continuou trabalhando até sua morte.

Cantor é considerado o fundador da teoria dos conjuntos. Suas contribuições nessa área incluem a descoberta de que o conjunto dos números reais é não numerável, ou seja, incontável. Ele também é lembrado por suas importantes contribuições em análise. Cantor também se interessava por filosofia e escreveu muitos artigos relacionando sua teoria dos conjuntos à metafísica.

Cantor casou-se em 1874 e teve cinco filhos. Seu temperamento melancólico foi equilibrado pela disposição e felicidade de sua mulher. Embora tenha recebido uma grande herança de seu pai, ele era mal pago como professor. Para suavizar isso, ele tentou obter um cargo com melhor remuneração na Universidade de Berlim. Sua indicação foi barrada por Kronecker, que não concordava com os pensamentos de Cantor sobre a teoria dos conjuntos. Cantor sofreu de uma doença mental nos últimos anos de sua vida. Ele morreu em 1918, em uma clínica psiquiátrica.

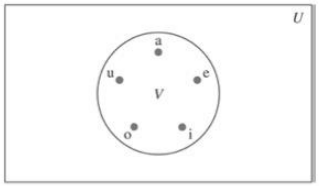


FIGURA 1 **Diagrama de Venn para o Conjunto de Vogais.**

EXEMPLO 7 Desenhe um diagrama de Venn que represente V, o conjunto de vogais do alfabeto da língua inglesa.

Solução: Desenhamos um retângulo para indicar o conjunto universo U, que é o conjunto das 26 letras do alfabeto da língua inglesa. Dentro desse retângulo, desenhamos um círculo para representar V. Dentro desse círculo, indicamos os elementos de V com pontos (veja a Figura 1). ◀

Há um conjunto especial que não tem elementos. Esse conjunto é chamado de **conjunto vazio,** ou **conjunto nulo**, e indicado por $\emptyset$. O conjunto vazio também pode ser indicado por $\{\ \}$ (ou seja, representamos o conjunto vazio com um par de chaves que demarcam todos os elementos desse conjunto). Muitas vezes, um conjunto de elementos com certas propriedades revela-se o conjunto nulo. Por exemplo, o conjunto de todos os números inteiros positivos que são maiores que os seus quadrados é um conjunto nulo. Um conjunto com um único elemento é chamado de **conjunto unitário.**

Um erro comum é confundir o conjunto vazio $\emptyset$ com o conjunto $\{\emptyset\}$, que é um conjunto unitário. O único elemento do conjunto $\{\emptyset\}$ é o conjunto vazio por si próprio! Uma analogia útil para lembrar dessa diferença é pensar em pastas do sistema de arquivos de um computador. O conjunto vazio pode ser entendido como uma pasta vazia e o conjunto que contém apenas o conjunto vazio pode ser pensado como uma pasta com exatamente uma pasta dentro, neste caso, a pasta vazia.

DEFINIÇÃO 4 O conjunto A é um *subconjunto* de B se e somente se todo elemento de A for também um elemento de B. Usamos a notação $A \subseteq B$ para indicar que A é um subconjunto do conjunto B.

Vemos que $A \subseteq B$ se e somente se a quantificação

$$\forall x (x \in A \rightarrow x \in B)$$

for verdadeira.

Links

BERTRAND RUSSELL (1872–1970) Bertrand Russell nasceu em uma proeminente família inglesa ativa no movimento progressivo que tinha um grande comprometimento com a liberdade. Ele tornou-se órfão jovem e ficou sob os cuidados dos seus avós paternos, que o educaram em casa. Entrou para a Trinity College, em Cambridge, em 1890, onde se tornou excelente em matemática e ciências morais. Ele ganhou uma bolsa de estudos com base em seu trabalho sobre os fundamentos de geometria. Em 1910, a Trinity College nomeou-o professor em lógica e filosofia da matemática. Russell batalhou pelas causas progressivas ao longo de sua vida. Ele mantinha grandes visões pacifistas e seus protestos contra a Segunda Guerra foram causa de sua demissão da Trinity College. Ele ficou preso por 6 meses em 1918, por causa de um artigo que escreveu e que foi taxado de aliciante. Russell lutou pelo direito do voto feminino na Grã-Bretanha. Em 1961, com 89 anos, ele foi preso pela segunda vez pelos protestos a favor do desarmamento nuclear.

O maior trabalho de Russell foi o desenvolvimento dos princípios que poderiam ser usados como fundamentos de toda a matemática. Seu trabalho mais famoso é o *Principia Mathematica,* escrito com Alfred North Whitehead, que tentou deduzir toda a matemática usando um conjunto de axiomas primitivos. Ele escreveu muitos livros sobre filosofia, física e idéias políticas. Russell ganhou o Prêmio Nobel de Literatura em 1950.

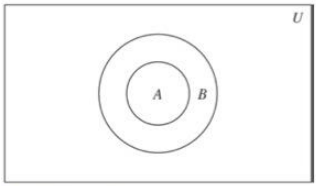


FIGURA 2 **Diagrama de Venn que mostra que A é um Subconjunto de B.**

EXEMPLO 8 O conjunto de todos os números inteiros positivos ímpares menores que 10 é um subconjunto do conjunto de todos os números inteiros positivos menores que 10, o conjunto de números racionais é um subconjunto do conjunto de números reais, o conjunto de todos os graduandos de ciência da computação em sua escola é um subconjunto do conjunto de todos os estudantes de sua escola e o conjunto de todas as pessoas da China é um subconjunto do conjunto de todas as pessoas da China (ou seja, é um subconjunto de si próprio). ◀

O Teorema 1 mostra que todo conjunto não vazio S tem, pelo menos, dois subconjuntos, o conjunto vazio e o próprio conjunto S, ou seja, $\emptyset \subseteq S$ e $S \subseteq S$.

TEOREMA 1

Para todo conjunto S,
(*i*) $\emptyset \subseteq S$ e (*ii*) $S \subseteq S$.

Demonstração: Demonstraremos (*i*) e deixaremos a demonstração de (*ii*) como um exercício.

Considere S como um conjunto. Para mostrar que $\emptyset \subseteq S$, devemos mostrar que $\forall x(x \in \emptyset \rightarrow x \in S)$ é verdadeira. Como o conjunto vazio não contém elementos, temos que $x \in \emptyset$ é sempre falsa. Temos que a proposição condicional $x \in \emptyset \rightarrow x \in S$ é sempre verdadeira, pois sua hipótese é sempre falsa e uma proposição condicional com uma hipótese falsa é verdadeira. Ou seja, $\forall x(x \in \emptyset \rightarrow x \in S)$ é verdadeira. Isto completa a demonstração de (*i*). Note que isto é um exemplo de demonstração por vacuidade. ◁

Quando desejamos enfatizar que o conjunto A é um subconjunto do conjunto B, mas que $A \neq B$, escrevemos $A \subset B$ e dizemos que A é um **subconjunto estrito** de B. Para $A \subset B$ ser verdadeira, deve ser o caso de $A \subseteq B$ e deve existir um elemento x de B que não é um elemento de A. Ou seja, A é um subconjunto estrito de B se

$$\forall x(x \in A \rightarrow x \in B) \wedge \exists x(x \in B \wedge x \notin A)$$

for verdadeira. Os diagramas de Venn podem ser usados para ilustrar que um conjunto A é um subconjunto de um conjunto B. Desenhamos o conjunto universo U como um retângulo. Dentro desse retângulo, desenhamos um círculo para B. Como A é um subconjunto de B, desenhamos um círculo para A dentro do círculo para B. Essa relação é mostrada na Figura 2.

Links

JOHN VENN (1834–1923) John Venn nasceu em uma família do subúrbio de Londres conhecida por sua filantropia. Ele freqüentou escolas londrinas e conquistou seu diploma em matemática na Caius College, em Cambridge, em 1857. Foi eleito membro de sua faculdade e manteve esse cargo até sua morte. Ele freqüentou o seminário em 1859 e, depois de uma breve carreira religiosa, retornou a Cambridge, onde desenvolveu trabalhos na área de ciência moral. Além de seu trabalho em matemática, Venn tinha interesse em história, e escreveu intensamente sobre sua faculdade e sua família.

O livro de Venn, *Lógica Simbólica*, torna claras algumas idéias apresentadas originalmente por Boole. Nesse livro, Venn apresenta um desenvolvimento sistemático de um método que usa figuras geométricas, conhecido agora como *diagrama de Venn*. Hoje, esses diagramas são usados principalmente para analisar argumentos lógicos e ilustrar relações entre conjuntos. Além disso, com seus trabalhos em lógica simbólica, Venn fez contribuições para a teoria da probabilidade, descritas em seu livro teórico largamente usado nesta área.

Uma maneira de mostrar que dois conjuntos têm os mesmos elementos é mostrar que cada conjunto é um subconjunto do outro. Em outras palavras, podemos mostrar que se A e B são conjuntos com $A \subseteq B$ e $B \subseteq A$, então $A = B$. Isto se torna uma maneira útil de mostrar que dois conjuntos são iguais. Ou seja, $A = B$, em que A e B são conjuntos, se e somente se $\forall x(x \in A \rightarrow x \in B)$ e $\forall x(x \in B \rightarrow x \in A)$, ou de modo análogo, se e somente se $\forall x(x \in A \leftrightarrow x \in B)$.

Os conjuntos podem ter outros conjuntos como elementos. Por exemplo, temos os conjuntos

$$A = \{\emptyset, \{a\}, \{b\}, \{a, b\}\} \text{ e } B = \{x \mid x \text{ é um subconjunto do conjunto } \{a, b\}\}.$$

Note que esses dois conjuntos são iguais, ou seja, $A = B$. Note também que $\{a\} \in A$, mas $a \notin A$.

Usamos conjuntos em muitos problemas de contagem e, para tais aplicações, precisamos discutir o tamanho dos conjuntos.

DEFINIÇÃO 5 Considere S como um conjunto. Se há exatamente n elementos distintos em S, em que n é um número inteiro não negativo, dizemos que S é um *conjunto finito* e que n é *cardinal* de S. O cardinal de S é indicado por $|S|$.

EXEMPLO 9 Considere A como o conjunto de números inteiros positivos ímpares menores que 10. Então, $|A| = 5$. ◀

EXEMPLO 10 Considere S como o conjunto de letras do alfabeto da língua inglesa. Então, $|S| = 26$. ◀

EXEMPLO 11 Como o conjunto nulo não tem elementos, temos que $|\emptyset| = 0$. ◀

Temos também interesse em conjuntos que não são finitos.

DEFINIÇÃO 6 Um conjunto é dito *infinito* se ele não é finito.

EXEMPLO 12 O conjunto de números inteiros positivos é infinito. ◀

Exemplos Extras

A cardinalidade dos conjuntos infinitos será discutida na Seção 2.4.

O Conjunto de Partes

Muitos problemas envolvem teste de todas as combinações de elementos de um conjunto para ver se eles satisfazem alguma propriedade. Para considerar todas as combinações de elementos de um conjunto S, construímos um novo conjunto que tem como elementos todos os subconjuntos de S.

DEFINIÇÃO 7 Dado um conjunto S, o *conjunto das partes* de S é o conjunto de todos os subconjuntos do conjunto S. O conjunto de partes de S é indicado por $P(S)$.

EXEMPLO 13 Qual o conjunto das partes do conjunto $\{0, 1, 2\}$?

Exemplos Extras

Solução: O conjunto das partes $P(\{0, 1, 2\})$ é o conjunto de todos os subconjuntos de $\{0, 1, 2\}$. Portanto,

$$P(\{0, 1, 2\}) = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}.$$

Note que o conjunto vazio e o próprio conjunto são elementos desse conjunto de subconjuntos. ◀

EXEMPLO 14 Qual o conjunto das partes do conjunto vazio? Qual o conjunto das partes do conjunto $\{\emptyset\}$?

Solução: O conjunto vazio tem exatamente um subconjunto, ou seja, ele mesmo. Conseqüentemente,

$$P(\emptyset) = \{\emptyset\}.$$

O conjunto $\{\emptyset\}$ tem exatamente dois subconjuntos, ou seja, $\emptyset$ e o próprio conjunto $\{\emptyset\}$. Assim,

$$P(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}.$$ ◀

Se um conjunto tem n elementos, então o seu conjunto de partes tem 2^n elementos. Demonstraremos esse fato de várias maneiras em seções subseqüentes do texto.

Produto Cartesiano

Freqüentemente, a ordem dos elementos em uma coleção é importante. Como os conjuntos não são ordenados, é necessária uma estrutura diferente para representar coleções ordenadas. Isso é determinado pela ***n*-upla ordenada.**

DEFINIÇÃO 8 A *n-upla ordenada* $(a_1, a_2, \ldots, a_n)$ é a coleção ordenada que tem a_1 como seu primeiro elemento, a_2 como seu segundo elemento, ... e a_n como seu n-ésimo elemento.

Dizemos que duas n-uplas ordenadas são iguais se e somente se cada par correspondente de seus elementos é igual. Em outras palavras, $(a_1, a_2, \ldots, a_n) = (b_1, b_2, \ldots, b_n)$ se e somente se $a_i = b_i$, para $i = 1, 2, \ldots, n$. Em particular, 2-uplas são chamadas de **pares ordenados.** Os pares ordenados (a, b) e (c, d) são iguais se e somente se $a = c$ e $b = d$. Note que (a, b) e (b, a) não são iguais, a menos que $a = b$.

Muitas das estruturas discretas que estudaremos nos próximos capítulos são baseadas na noção do *produto cartesiano* dos conjuntos (assim chamado depois de René Descartes). Primeiro definiremos o produto cartesiano de dois conjuntos.

Links

RENÉ DESCARTES (1596–1650) René Descartes nasceu em uma família nobre perto de Tours, na França, aproximadamente 200 milhas a sudoeste de Paris. Ele foi o terceiro filho da primeira esposa de seu pai; ela morreu alguns dias após seu nascimento. Por causa da frágil saúde de René, seu pai, um juiz provinciano, abdicou das lições formais de seu filho até os 8 anos, quando René entrou para um colégio jesuíta em La Flèche. Em conseqüência de seu estado de saúde, o reitor da escola permitia que ele permanecesse descansando até mais tarde. A partir daí, Descartes passava suas manhãs na cama; ele considerava esse tempo como as suas horas mais produtivas de pensamento.

Descartes deixou a escola em 1612, mudando-se para Paris, onde passou 2 anos estudando matemática. Ele obteve um diploma de direito em 1616, na Universidade de Poitiers. Aos 18 anos, Descartes desanimou com os estudos e decidiu conhecer o mundo. Mudou-se para Paris e tornou-se um apostador de sucesso. Entretanto, ele se cansou da vida rude e mudou-se para o subúrbio de Saint-Germain, onde devotou sua vida aos estudos matemáticos. Quando seus amigos apostadores o encontraram, decidiu sair da França e entrar para a carreira militar. Entretanto, ele nunca foi um combatente. Um dia, enquanto se protegia do frio em uma sala aquecida no acampamento militar, ele teve muitas alucinações, as quais revelaram sua carreira futura como um matemático e filósofo.

Depois de terminar sua carreira militar, ele viajou pela Europa. Em seguida, passou alguns anos em Paris, onde estudou matemática e filosofia e construiu instrumentos ópticos. Descartes decidiu se mudar para a Holanda, onde passou 20 anos percorrendo todo o país, desenvolvendo seu mais importante trabalho. Durante esse tempo, ele escreveu muitos livros, incluindo o *Discours,* que contém sua contribuição para a geometria analítica, área pela qual é mais conhecido. Fez também contribuições fundamentais para a filosofia.

Em 1649, Descartes foi convidado pela Rainha Christina para visitar sua corte na Suécia, como seu tutor em filosofia. Embora fosse relutante em viver no que ele chamava de "a terra de ursos entre rochas e gelo", ele finalmente aceitou seu convite e mudou-se para a Suécia. Infelizmente, o inverno de 1649–1650 foi extremamente intenso. Descartes pegou uma pneumonia e morreu em fevereiro.

DEFINIÇÃO 9 Considere A e B como conjuntos. O *produto cartesiano* de A e B, indicado por $A \times B$, é o conjunto de todos os pares ordenados (a, b), em que $a \in A$ e $b \in B$. Assim,

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}.$$

EXEMPLO 15 Considere A como o conjunto de todos os estudantes em uma universidade e B, como o conjunto de todos os cursos oferecidos por essa universidade. Qual o produto cartesiano de $A \times B$?

Exemplos Extras

Solução: O produto cartesiano de $A \times B$ consiste em todos os pares ordenados de forma (a, b), em que a é um estudante da universidade e b, um curso oferecido por essa universidade. O conjunto $A \times B$ pode ser usado para representar todas as possíveis matrículas em um curso da universidade. ◀

EXEMPLO 16 Qual o produto cartesiano de $A = \{1, 2\}$ e $B = \{a, b, c\}$?

Solução: O produto cartesiano de $A \times B$ é

$$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}.$$ ◀

Um subconjunto R do produto cartesiano $A \times B$ é chamado de **relação** do conjunto A com o conjunto B. Os elementos de R são pares ordenados, em que o primeiro elemento pertence a A e o segundo, a B. Por exemplo, $R = \{(a, 0), (a, 1), (a, 3), (b, 1), (b, 2), (c, 0), (c, 3)\}$ é uma relação do conjunto $\{a, b, c\}$ com o conjunto $\{0, 1, 2, 3\}$. Estudaremos as relações mais a fundo no Capítulo 8.

Os produtos cartesianos $A \times B$ e $B \times A$ não são iguais, a menos que $A = \emptyset$ ou $B = \emptyset$ (de modo que $A \times B = \emptyset$) ou $A = B$ (veja os exercícios 26 e 30, no final desta seção). Isto é ilustrado no Exemplo 17.

EXEMPLO 17 Mostre que o produto cartesiano $B \times A$ não é igual ao produto cartesiano $A \times B$, em que A e B são os conjuntos do Exemplo 16.

Solução: O produto cartesiano $B \times A$ é

$$B \times A = \{(a, 1), (a, 2), (b, 1), (b, 2), (c, 1), (c, 2)\}.$$

Isso não é igual a $A \times B$, que foi encontrado no Exemplo 16. ◀

O produto cartesiano de mais de dois conjuntos também pode ser definido.

DEFINIÇÃO 10 O *produto cartesiano* dos conjuntos $A_1, A_2, \ldots, A_n$, indicado por $A_1 \times A_2 \times \ldots \times A_n$, é o conjunto de n-uplas ordenadas $(a_1, a_2, \ldots, a_n)$, em que a_i pertence a A_i para $i = 1, 2, \ldots, n$. Em outras palavras,

$$A_1 \times A_2 \times \ldots \times A_n = \{(a_1, a_2, \ldots, a_n) \mid a_i \in A_i \text{ para } i = 1, 2, \ldots, n\}.$$

EXEMPLO 18 Qual é o produto cartesiano de $A \times B \times C$, em que $A = \{0, 1\}$, $B = \{1, 2\}$ e $C = \{0, 1, 2\}$?

Solução: O produto cartesiano de $A \times B \times C$ consiste em todos os trios ordenados (a, b, c), em que $a \in A$, $b \in B$ e $c \in C$. Assim,

$$A \times B \times C = \{(0, 1, 0), (0, 1, 1), (0, 1, 2), (0, 2, 0), (0, 2, 1), (0, 2, 2), (1, 1, 0), (1, 1, 1), (1, 1, 2), (1, 2, 0), (1, 2, 1), (1, 2, 2)\}.$$ ◀

Usando Notação de Conjuntos com Quantificadores

Às vezes, restringimos o domínio de uma sentença determinada explicitamente fazendo uso de uma notação específica. Por exemplo, $\forall x \in S(P(x))$ indica a quantificação universal de $P(x)$ de todos os elementos do conjunto S. Em outras palavras, $\forall x \in S(P(x))$ representa $\forall x(x \in S \rightarrow P(x))$. Semelhantemente, $\exists x \in S(P(x))$ indica a quantificação existencial de $P(x)$ sobre os elementos em S. Ou seja, $\exists x \in S(P(x))$ representa $\exists x(x \in S \wedge P(x))$.

EXEMPLO 19 Qual o significado das proposições $\forall x \in \mathbf{R}\,(x^2 \geq 0)$ e $\exists x \in \mathbf{Z}\,(x^2 = 1)$?

Solução: A proposição $\forall x \in \mathbf{R}\,(x^2 \geq 0)$ afirma que para todo número real x, $x^2 \geq 0$. Essa proposição pode ser expressa como "O quadrado de todo número real é não negativo." Essa é uma proposição verdadeira.

A proposição $\exists x \in \mathbf{Z}(x^2 = 1)$ afirma que existe um número inteiro x, tal que $x^2 = 1$. Essa proposição pode ser expressa como "Há um número inteiro cujo quadrado é 1". Isso também é verdadeiro, pois $x = 1$ é tal número inteiro (assim como -1). ◀

Conjuntos-Verdade de Quantificadores

Vamos agora combinar conceitos da teoria dos conjuntos com os de predicados lógicos. Dados um predicado P e um domínio D, definimos o **conjunto-verdade** de P como o conjunto de elementos x em D para que $P(x)$ seja verdadeira. O conjunto-verdade de $P(x)$ é indicado por $\{x \in D \mid P(x)\}$.

EXEMPLO 20 Quais são os conjuntos-verdade dos predicados $P(x)$, $Q(x)$ e $R(x)$, em que o domínio é o conjunto dos números inteiros e $P(x)$ é "$|x| = 1$," $Q(x)$ é "$x^2 = 2$" e $R(x)$ é "$|x| = x$"?

Solução: O conjunto-verdade de P, $\{x \in \mathbf{Z} \mid |x| = 1\}$, é o conjunto dos números inteiros com $|x| = 1$. Como $|x| = 1$ quando $x = 1$ ou $x = -1$, e para nenhum outro número inteiro x, vemos que o conjunto-verdade de P é o conjunto $\{-1, 1\}$.

O conjunto-verdade de Q, $\{x \in \mathbf{Z} \mid x^2 = 2\}$, é o conjunto dos números inteiros com $x^2 = 2$. Esse é um conjunto vazio, pois não há nenhum número inteiro x para que $x^2 = 2$.

O conjunto-verdade de R, $\{x \in \mathbf{Z} \mid |x| = x\}$, é o conjunto dos números inteiros com $|x| = x$. Como $|x| = x$ se e somente se $x \geq 0$, temos que o conjunto-verdade R é $\mathbf{N}$, o conjunto dos números inteiros não negativos. ◀

Note que $\forall x\, P(x)$ é verdadeira sobre o domínio U se e somente se o conjunto-verdade de P for o conjunto U. Da mesma forma, $\exists x\, P(x)$ é verdadeira sobre o domínio U se e somente se o conjunto-verdade de P não for vazio.

Exercícios

1. Liste todos os elementos dos conjuntos abaixo.
- **a)** $\{x \mid x$ é um número real, tal que $x^2 = 1\}$
- **b)** $\{x \mid x$ é um número inteiro positivo menor que $12\}$
- **c)** $\{x \mid x$ é o quadrado de um número inteiro e $x < 100\}$
- **d)** $\{x \mid x$ é um número inteiro, tal que $x^2 = 2\}$

2. Use a notação de construção de conjuntos para dar uma descrição de cada um dos conjuntos abaixo.
- **a)** $\{0, 3, 6, 9, 12\}$
- **b)** $\{-3, -2, -1, 0, 1, 2, 3\}$
- **c)** $\{m, n, o, p\}$

3. Determine se cada um dos pares dos conjuntos a seguir são iguais.
- **a)** $\{1, 3, 3, 3, 5, 5, 5, 5, 5\}$, $\{5, 3, 1\}$
- **b)** $\{\{1\}\}$, $\{1, \{1\}\}$
- **c)** $\emptyset$, $\{\emptyset\}$

4. Suponha que $A = \{2, 4, 6\}$, $B = \{2, 6\}$, $C = \{4, 6\}$ e $D = \{4, 6, 8\}$. Determine quais desses conjuntos são subconjuntos de outro desses conjuntos.

5. Para cada um dos conjuntos abaixo, determine se 2 é um elemento do conjunto.
- **a)** $\{x \in \mathbf{R} \mid x$ é um número inteiro maior que $1\}$
- **b)** $\{x \in \mathbf{R} \mid x$ é o quadrado de um número inteiro$\}$
- **c)** $\{2, \{2\}\}$
- **d)** $\{\{2\}, \{\{2\}\}\}$
- **e)** $\{\{2\}, \{2, \{2\}\}\}$
- **f)** $\{\{\{2\}\}\}$

6. Para cada um dos conjuntos do Exercício 5, determine se $\{2\}$ é um elemento do conjunto.

7. Determine se cada uma das proposições abaixo é verdadeira ou falsa.
a) $0 \in \emptyset$ **b)** $\emptyset \in \{0\}$
c) $\{0\} \subset \emptyset$ **d)** $\emptyset \subset \{0\}$
e) $\{0\} \in \{0\}$ **f)** $\{0\} \subset \{0\}$
g) $\{\emptyset\} \subseteq \{\emptyset\}$

8. Determine se cada uma das proposições abaixo é verdadeira ou falsa.
a) $\emptyset \in \{\emptyset\}$ **b)** $\emptyset \in \{\emptyset, \{\emptyset\}\}$
c) $\{\emptyset\} \in \{\emptyset\}$ **d)** $\{\emptyset\} \in \{\{\emptyset\}\}$
e) $\{\emptyset\} \subset \{\emptyset, \{\emptyset\}\}$ **f)** $\{\{\emptyset\}\} \subset \{\emptyset, \{\emptyset\}\}$
g) $\{\{\emptyset\}\} \subset \{\{\emptyset\}, \{\emptyset\}\}$

9. Determine se cada uma das proposições abaixo é verdadeira ou falsa.
a) $x \in \{x\}$ **b)** $\{x\} \subseteq \{x\}$ **c)** $\{x\} \in \{x\}$
d) $\{x\} \in \{\{x\}\}$ **e)** $\emptyset \subseteq \{x\}$ **f)** $\emptyset \in \{x\}$

10. Use um diagrama de Venn para ilustrar o subconjunto dos números inteiros ímpares no conjunto de todos os números inteiros positivos não excedentes a 10.

11. Use um diagrama de Venn para ilustrar o conjunto de todos os meses do ano cujos nomes não contêm a letra R no conjunto de todos os meses do ano.

12. Use um diagrama de Venn para ilustrar a relação $A \subseteq B$ e $B \subseteq C$.

13. Use um diagrama de Venn para ilustrar a relação $A \subset B$ e $B \subset C$.

14. Use um diagrama de Venn para ilustrar a relação $A \subset B$ e $A \subset C$.

15. Suponha que A, B e C sejam conjuntos, tal que $A \subseteq B$ e $B \subseteq C$. Mostre que $A \subseteq C$.

16. Encontre dois conjuntos A e B, tal que $A \in B$ e $A \subseteq B$.

17. Qual é a cardinalidade de cada um dos conjuntos abaixo?
a) $\{a\}$
b) $\{\{a\}\}$
c) $\{a, \{a\}\}$
d) $\{a, \{a\}, \{a, \{a\}\}\}$

18. Qual é a cardinalidade de cada um dos conjuntos abaixo?
a) $\emptyset$ **b)** $\{\emptyset\}$
c) $\{\emptyset, \{\emptyset\}\}$ **d)** $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$

19. Encontre o conjunto de partes para cada um dos conjuntos abaixo, em que a e b são elementos distintos.
a) $\{a\}$
b) $\{a, b\}$
c) $\{\emptyset, \{\emptyset\}\}$

20. Você pode concluir que $A = B$ se A e B são dois conjuntos com o mesmo conjunto de partes?

21. Quantos elementos cada um dos conjuntos abaixo têm, se a e b são elementos distintos?
a) $P(\{a, b, \{a, b\}\})$
b) $P(\{\emptyset, a, \{a\}, \{\{a\}\}\})$
c) $P(P(\emptyset))$

22. Determine se cada um dos conjuntos abaixo é o conjunto de partes de um conjunto, em que a e b são elementos distintos.
a) $\emptyset$ **b)** $\{\emptyset, \{a\}\}$
c) $\{\emptyset, \{a\}, \{\emptyset, a\}\}$ **d)** $\{\emptyset, \{a\}, \{b\}, \{a, b\}\}$

23. Considere $A = \{a, b, c, d\}$ e $B = \{y, z\}$. Encontre
a) $A \times B$ **b)** $B \times A$

24. Qual o produto cartesiano de $A \times B$, em que A é o conjunto de cursos oferecidos pelo departamento de matemática em uma universidade e B, o conjunto de professores de matemática nessa universidade?

25. Qual o produto cartesiano de $A \times B \times C$, em que A é o conjunto de todas as empresas aéreas e B e C são o conjunto de todas as cidades dos Estados Unidos?

26. Suponha que $A \times B = \emptyset$, em que A e B são conjuntos. O que você pode concluir?

27. Considere A como um conjunto. Mostre que $\emptyset \times A = A \times \emptyset = \emptyset$.

28. Considere $A = \{a, b, c\}$, $B = \{x, y\}$ e $C = \{0, 1\}$. Encontre
a) $A \times B \times C$ **b)** $C \times B \times A$
c) $C \times A \times B$ **d)** $B \times B \times B$

29. Quantos elementos diferentes $A \times B$ tem, se A tem m elementos e B, n elementos?

30. Mostre que $A \times B \neq B \times A$, em que A e B são conjuntos não vazios, a menos que $A = B$.

31. Explique por que $A \times B \times C$ e $(A \times B) \times C$ não são iguais.

32. Explique por que $(A \times B) \times (C \times D)$ e $A \times (B \times C) \times D$ não são iguais.

33. Transcreva cada uma das quantificações abaixo em português e determine seu valor-verdade.
a) $\forall x \in \mathbf{R}\ (x^2 \neq -1)$ **b)** $\exists x \in \mathbf{Z}\ (x^2 = 2)$
c) $\forall x \in \mathbf{Z}\ (x^2 > 0)$ **d)** $\exists x \in \mathbf{R}\ (x^2 = x)$

34. Transcreva cada uma das quantificações abaixo em português e determine seu valor-verdade.
a) $\exists x \in \mathbf{R}\ (x^3 = -1)$ **b)** $\exists x \in \mathbf{Z}\ (x + 1 > x)$
c) $\forall x \in \mathbf{Z}\ (x - 1 \in \mathbf{Z})$ **d)** $\forall x \in \mathbf{Z}\ (x^2 \in \mathbf{Z})$

35. Encontre o conjunto-verdade de cada um dos predicados abaixo, em que o domínio é o conjunto dos números inteiros.
a) $P(x)$: "$x^2 < 3$"
b) $Q(x)$: "$x^2 > x$"
c) $R(x)$: "$2x + 1 = 0$"

36. Encontre o conjunto-verdade de cada um dos predicados abaixo, em que o domínio é o conjunto dos números inteiros.
a) $P(x)$: "$x^3 \geq 1$"
b) $Q(x)$: "$x^2 = 2$"
c) $R(x)$: "$x < x^2$"

***37.** Para pares ordenados serem bem definidos, precisamos da propriedade de igualdade que diz que dois pares ordenados são iguais se e somente se os primeiros elementos dos pares forem iguais e os segundos elementos também. Surpreendentemente, em vez de o par ordenado ser tomado como um conceito primitivo, podemos construir pares ordenados usando noções básicas da teoria dos conjuntos. Mostre que se nós definirmos o par ordenado (a, b) como $\{\{a\}, \{a, b\}\}$, então $(a, b) = (c, d)$ se e somente se $a = c$ e $b = d$. [*Dica:*

Primeiro mostre que $\{\{a\}, \{a, b\}\} = \{\{c\}, \{c, d\}\}$ se e somente se $a = c$ e $b = d$.]

***38.** Este exercício apresenta o **paradoxo de Russell**. Considere S como o conjunto que contém um conjunto x se o conjunto x não pertencer a ele mesmo, para que $S = \{x \mid x \notin x\}$.

a) Mostre a hipótese de que S é um elemento de S leva a uma contradição.

b) Mostre a hipótese de que S não é um elemento de S leva a uma contradição.

Pelos itens (a) e (b), temos que o conjunto S não pode ser definido como foi feito. Esse paradoxo pode ser evitado restringindo os tipos de elementos que os conjuntos podem ter.

***39.** Descreva um procedimento para listar todos os subconjuntos de um conjunto finito.

2.2 Operações com Conjuntos

Introdução

Dois conjuntos podem ser combinados de muitas maneiras diferentes. Por exemplo, começando com o conjunto dos graduandos em matemática e o conjunto dos graduandos em ciência da computação, ambos da sua faculdade, podemos formar o conjunto dos estudantes que são ou graduandos em matemática ou em ciência da computação, o conjunto dos estudantes que são da matemática e da ciência da computação, o conjunto de todos os estudantes que não são graduandos em matemática, e assim por diante.

Links

DEFINIÇÃO 1 Sejam A e B conjuntos. A *união* dos conjuntos A e B, indicada por $A \cup B$, é o conjunto que contém aqueles elementos que estão em A ou em B, ou em ambos.

Um elemento x pertence à união dos conjuntos A e B se e somente se x pertencer a A ou x pertencer a B. Isto nos mostra que

$$A \cup B = \{x \mid x \in A \vee x \in B\}.$$

O diagrama de Venn mostrado na Figura 1 representa a união de dois conjuntos A e B. A área que representa $A \cup B$ é a área sombreada dos círculos que representam ou A ou B.

Daremos alguns exemplos de união de conjuntos.

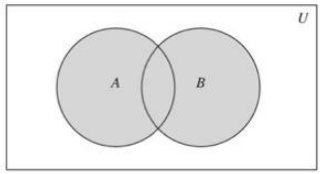
EXEMPLO 1 A união dos conjuntos $\{1, 3, 5\}$ e $\{1, 2, 3\}$ é o conjunto $\{1, 2, 3, 5\}$; ou seja, $\{1, 3, 5\} \cup \{1, 2, 3\} = \{1, 2, 3, 5\}$. ◀

EXEMPLO 2 A união do conjunto de todos os estudantes de ciência da computação com o conjunto de todos os estudantes de matemática da sua faculdade é o conjunto dos estudantes da sua faculdade que são graduandos ou em matemática ou em ciência da computação (ou em ambas). ◀

DEFINIÇÃO 2 Sejam A e B conjuntos. A *interseção* dos conjuntos A e B, indicada por $A \cap B$, é o conjunto que contém aqueles elementos que estão em A e em B, simultaneamente.

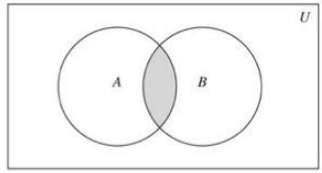
Um elemento x pertence à interseção dos conjuntos A e B se e somente se x pertencer a A e x pertencer a B. Isso nos mostra que

$$A \cap B = \{x \mid x \in A \wedge x \in B\}.$$



$A \cup B$ está sombreado.

FIGURA 1 Diagrama de Venn que Representa a União de A e B.



$A \cap B$ está sombreado.

FIGURA 2 Diagrama de Venn que Representa a Intersecção de A e B.

O diagrama de Venn mostrado na Figura 2 representa a intersecção de dois conjuntos A e B. A área sombreada, que está entre os dois círculos que representam os conjuntos A e B, é a área que representa a intersecção de A e B.

Daremos alguns exemplos de intersecção de conjuntos.

EXEMPLO 3 A intersecção dos conjuntos $\{1, 3, 5\}$ e $\{1, 2, 3\}$ é o conjunto $\{1, 3\}$; ou seja, $\{1, 3, 5\} \cap \{1, 2, 3\} = \{1, 3\}$. ◀

EXEMPLO 4 A intersecção do conjunto de todos os estudantes de ciência da computação com o conjunto de todos os estudantes de matemática de sua faculdade é o conjunto de todos os estudantes que fazem tanto ciência da computação quanto matemática. ◀

DEFINIÇÃO 3 Dois conjuntos são chamados de *disjuntos* se sua intersecção é um conjunto vazio.

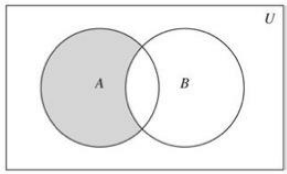
EXEMPLO 5 Considere $A = \{1, 3, 5, 7, 9\}$ e $B = \{2, 4, 6, 8, 10\}$. Como $A \cap B = \emptyset$, A *e* B são disjuntos. ◀

Freqüentemente temos interesse em encontrar a cardinalidade de uma união de dois conjuntos finitos A e B. Note que $|A| + |B|$ faz a contagem de cada elemento que está somente em A e cada elemento que está somente em B, isso faz com que cada elemento que está em ambos, A e B, seja contado duas vezes. Então, se o número de elementos que está em ambos os conjuntos A e B é subtraído de $|A| + |B|$, os elementos em $A \cap B$ serão contados apenas uma vez. Assim,

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

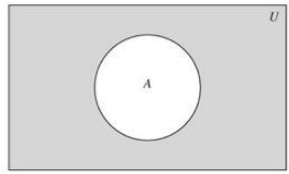
A generalização desse resultado de uniões de um número arbitrário de conjuntos é chamado de **princípio da inclusão-exclusão.** O princípio da inclusão-exclusão é uma técnica importante usada em enumeração. Discutiremos com detalhes esse princípio e outras técnicas de contagem nos capítulos 5 e 7.

Há outras maneiras importantes de combinar conjuntos.



$A - B$ está sombreado.

FIGURA 3 Diagrama de Venn para a Diferença entre A e B.



$\overline{A}$ está sombreado.

FIGURA 4 Diagrama de Venn para o Complemento do Conjunto A.

DEFINIÇÃO 4 Sejam A e B conjuntos. A *diferença* entre A e B, indicada por $A - B$, é o conjunto que contém aqueles elementos que estão em A mas não estão em B. A diferença entre A e B é também chamada de *complemento de B em relação a A*.

Um elemento x pertence à diferença de A e B se e somente se $x \in A$ e $x \notin B$. Isso nos mostra que

$$A - B = \{x \mid x \in A \wedge x \notin B\}.$$

O diagrama de Venn mostrado na Figura 3 representa a diferença entre os conjuntos A e B. A área sombreada dentro do círculo que representa A e fora do círculo que representa B é a área que representa $A - B$.

Daremos alguns exemplos de diferenças de conjuntos.

EXEMPLO 6 A diferença entre $\{1, 3, 5\}$ e $\{1, 2, 3\}$ é o conjunto $\{5\}$; ou seja, $\{1, 3, 5\} - \{1, 2, 3\} = \{5\}$. Isso não é o mesmo que a diferença entre $\{1, 2, 3\}$ e $\{1, 3, 5\}$, que é o conjunto $\{2\}$. ◀

EXEMPLO 7 A diferença entre o conjunto dos estudantes de ciência da computação e o conjunto dos estudantes de matemática da sua faculdade é o conjunto de todos os estudantes de ciência da computação que não são estudantes de matemática. ◀

Uma vez que o conjunto universo U foi especificado, o **complemento** de um conjunto pode ser definido.

DEFINIÇÃO 5 Considere U como o conjunto universo. O *complemento* do conjunto A, indicado por $\overline{A}$, é o complemento de A em relação à U. Em outras palavras, o complemento do conjunto A é $U - A$.

Um elemento pertence a $\overline{A}$ se e somente se $x \notin A$. Isso nos mostra que

$$\overline{A} = \{x \mid x \notin A\}.$$

Na Figura 4, a área sombreada fora do círculo que representa A é a área que representa $\overline{A}$.

Daremos alguns exemplos do complemento de um conjunto.

TABELA 1 Identidades de Conjuntos.

Identidade	*Nome*
$A \cup \emptyset = A$ $A \cap U = A$	Propriedades dos elementos neutros
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Propriedades de dominação
$A \cup A = A$ $A \cap A = A$	Propriedades idempotentes
$\overline{(\overline{A})} = A$	Propriedades da complementação
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Propriedades comutativas
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Propriedades associativas
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Propriedades distributivas
$\overline{A \cup B} = \overline{A} \cap \overline{B}$ $\overline{A \cap B} = \overline{A} \cup \overline{B}$	Leis de De Morgan
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Propriedades de absorção
$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$	Propriedades dos complementares

EXEMPLO 8 Considere $A = \{a, e, i, o, u\}$ (em que o conjunto universo é o conjunto de letras do alfabeto da língua inglesa). Então, $\overline{A} = \{b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z\}$. ◀

EXEMPLO 9 Considere A como o conjunto dos números inteiros positivos maiores que 10 (com o conjunto universo, o conjunto de todos os números inteiros positivos). Então, $\overline{A} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. ◀

Identidades de Conjuntos

A Tabela 1 lista as mais importantes identidades de conjuntos. Demonstraremos muitas dessas identidades aqui, usando três métodos diferentes. Esses métodos são apresentados para ilustrar que geralmente há muitas maneiras diferentes para solucionar um problema. As demonstrações das identidades restantes serão deixadas como exercícios. O leitor deve notar a semelhança entre essas identidades e as equivalências lógicas discutidas na Seção 1.2. De fato, as identidades de conjuntos podem ser demonstradas diretamente a partir da correspondência de equivalências lógicas. Além disso, ambos são casos especiais de identidades firmadas pela álgebra booleana (discutida no Capítulo 11).

Uma maneira de mostrar que dois conjuntos são iguais é mostrar que cada um é um subconjunto do outro. Lembre-se de que mostrar que um conjunto é um subconjunto de um segundo conjunto é mostrar que se um elemento pertence ao primeiro conjunto, então ele deverá pertencer também ao segundo conjunto. Usamos geralmente uma demonstração direta para fazer isso. Ilustraremos esse tipo de demonstração para estabelecer a segunda lei de De Morgan.

EXEMPLO 10 Demonstre que $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

Solução: Para demonstrar que $\overline{A \cap B} = \overline{A} \cup \overline{B}$, mostraremos que $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$ e que $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$.

Exemplos Extras

Primeiro, mostraremos que $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$. Então, suponha que $x \in \overline{A \cap B}$. Pela definição de complemento, $x \notin A \cap B$. Pela definição de intersecção, $\neg ((x \in A) \wedge (x \in B))$ é verdadeira. Aplicando a lei de De Morgan (da lógica), vemos que $\neg(x \in A)$ ou $\neg (x \in B)$. Assim, pela definição de negação, $x \notin A$ ou $x \notin B$. Pela definição de complemento, $x \in \overline{A}$ ou $x \in \overline{B}$. Temos que, pela definição de união, $x \in \overline{A} \cup \overline{B}$. Isso mostra que $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$.

Depois, mostraremos que $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$. Suponha agora que $x \in \overline{A} \cup \overline{B}$. Pela definição de união, $x \in \overline{A}$ ou $x \in \overline{B}$. Usando a definição de complemento, temos que $x \notin A$ ou $x \notin B$. Conseqüentemente, $\neg(x \in A) \vee \neg (x \in B)$ é verdadeira. Pela lei de De Morgan (da lógica), concluímos que $\neg((x \in A) \wedge (x \in B))$ é verdadeira. Pela definição de intersecção, temos que $\neg(x \in A \cap B)$ mantém-se. Usamos a definição de complemento para concluir que $x \in \overline{A \cap B}$. Isso mostra que $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$. Como mostramos que cada conjunto é um subconjunto do outro, os dois conjuntos são iguais e a identidade está demonstrada. ◀

Podemos expressar mais sucintamente a razão usada no Exemplo 10 usando a notação de construção do conjunto, como mostra o Exemplo 11.

EXEMPLO 11 Use a notação de construção do conjunto e equivalências lógicas para estabelecer a segunda lei de De Morgan $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

Solução: Podemos demonstrar essa identidade seguindo os seguintes passos.

$\overline{A \cap B} = \{x \mid x \notin A \cap B\}$	pela definição de complemento
$= \{x \mid \neg(x \in (A \cap B))\}$	pela definição do símbolo: não pertence
$= \{x \mid \neg(x \in A \wedge x \in B)\}$	pela definição de intersecção
$= \{x \mid \neg(x \in A) \vee \neg(x \in B)\}$	pela primeira lei de De Morgan para equivalências lógicas
$= \{x \mid x \notin A \vee x \notin B\}$	pela definição do símbolo: não pertence
$= \{x \mid x \in \overline{A} \vee x \in \overline{B}\}$	pela definição de complemento
$= \{x \mid x \in \overline{A} \cup \overline{B}\}$	pela definição de união
$= \overline{A} \cup \overline{B}$	pelo significado da notação de construção do conjunto

Note que por trás das definições de complemento, união, relações de conjunto e notação de construção do conjunto, essa demonstração utiliza a primeira lei de De Morgan para equivalências lógicas. ◀

Demonstrar a identidade de conjuntos que envolvem mais de dois conjuntos pelo método que requer mostrar que cada lado da identidade é um subconjunto do outro, normalmente requer que consideremos caminhos com casos diferentes, como ilustrado pela demonstração do Exemplo 12, uma das propriedades distributivas para conjuntos.

EXEMPLO 12 Demonstre a primeira propriedade distributiva da Tabela 1, que afirma que $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ para todos os conjuntos A, B e C.

Solução: Demonstraremos essa identidade mostrando que cada lado é um subconjunto do outro lado.

Suponha que $x \in A \cap (B \cup C)$. Então, $x \in A$ e $x \in B \cup C$. Pela definição de união, temos que $x \in A$ e $x \in B$ ou $x \in C$ (ou ambos). Conseqüentemente, sabemos que $x \in A$ e $x \in B$ ou que $x \in A$ e $x \in C$. Pela definição de intersecção, temos que $x \in A \cap B$ ou $x \in A \cap C$. Usando a definição de união, concluímos que $x \in (A \cap B) \cup (A \cap C)$. Concluímos que $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$.

TABELA 2 Uma Tabela de Pertinência para a Propriedade Distributiva.

A	B	C	$B \cup C$	$A \cap (B \cup C)$	$A \cap B$	$A \cap C$	$(A \cap B) \cup (A \cap C)$
1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1
1	0	1	1	1	0	1	1
1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0

Agora suponha que $x \in (A \cap B) \cup (A \cap C)$. Então, pela definição de união, $x \in A \cap B$ ou $x \in A \cap C$. Pela definição de intersecção, temos que $x \in A$ e $x \in B$ ou que $x \in A$ e $x \in C$. A partir disso, vemos que $x \in A$ e $x \in B$ ou $x \in C$. Conseqüentemente, pela definição de união, vemos que $x \in A$ e $x \in B \cup C$. Além disso, pela definição de intersecção, temos que $x \in A \cap (B \cup C)$. Concluímos que $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$. Isto completa a demonstração de identidade. ◀

As identidades de conjunto podem também ser demonstradas usando **tabelas de pertinência.** Consideremos cada combinação de pertinência de elementos em conjuntos e verificamos se os elementos pertencem a ambos os conjuntos na identidade. Para indicar que um elemento está em um conjunto, um 1 é usado; para indicar que um elemento não está no conjunto, um 0 é usado. (O leitor deve notar a semelhança entre as tabelas de pertinência e as tabelas-verdade.)

EXEMPLO 13 Use uma tabela de pertinência para mostrar que $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

Solução: A tabela de pertinência para essas combinações de conjuntos é mostrada na Tabela 2. Essa tabela tem oito linhas. Pelas colunas de $A \cap (B \cup C)$ e $(A \cap B) \cup (A \cap C)$ serem as mesmas, a identidade é válida. ◀

Identidades de conjuntos adicionais podem ser estabelecidas usando aquelas que já foram demonstradas. Considere o Exemplo 14.

EXEMPLO 14 Considere A, B e C como conjuntos. Mostre que

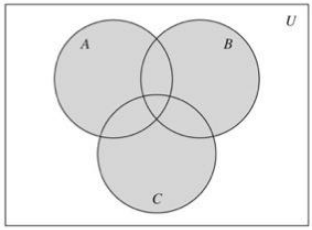
$$\overline{A \cup (B \cap C)} = (\overline{C} \cup \overline{B}) \cap \overline{A}.$$

Solução: Temos,

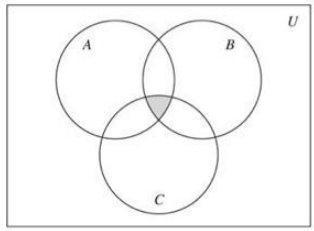
$$\begin{aligned}
\overline{A \cup (B \cap C)} &= \overline{A} \cap \overline{(B \cap C)} && \text{pela primeira lei de De Morgan} \\
&= \overline{A} \cap (\overline{B} \cup \overline{C}) && \text{pela segunda lei de De Morgan} \\
&= (\overline{B} \cup \overline{C}) \cap \overline{A} && \text{pela propriedade comutativa para intersecções} \\
&= (\overline{C} \cup \overline{B}) \cap \overline{A} && \text{pela propriedade comutativa para uniões.}
\end{aligned}$$ ◀

Uniões e Interseções Generalizadas

Como as uniões e interseções dos conjuntos satisfazem as propriedades associativas, os conjuntos $A \cup B \cup C$ e $A \cap B \cap C$ são bem definidos; ou seja, o significado dessa notação não é ambíguo quando A, B e C são conjuntos. Ou seja, não temos que usar parênteses para indicar



(a) $A \cup B \cup C$ está sombreado.



(b) $A \cap B \cap C$ está sombreado.

FIGURA 5 **A União e a Intersecção de *A*, *B* e *C*.**

qual operação é realizada primeiro, pois $A \cup (B \cup C) = (A \cup B) \cup C$ e $A \cap (B \cap C) = (A \cap B) \cap C$. Note que $A \cup B \cup C$ contém aqueles elementos que estão em, pelo menos, um dos conjuntos *A*, *B* e *C*, e que $A \cap B \cap C$ contém aqueles elementos que estão em todos os conjuntos *A*, *B* e *C*. Essas combinações dos três conjuntos, *A, B e C,* são mostradas na Figura 5.

EXEMPLO 15 Considere $A = \{0, 2, 4, 6, 8\}$, $B = \{0, 1, 2, 3, 4\}$ e $C = \{0, 3, 6, 9\}$. O que são $A \cup B \cup C$ *e* $A \cap B \cap C$?

Solução: O conjunto $A \cup B \cup C$ contém aqueles elementos que estão em, pelo menos, um dos conjuntos *A, B* e *C*. Assim,

$$A \cup B \cup C = \{0, 1, 2, 3, 4, 6, 8, 9\}.$$

O conjunto $A \cap B \cap C$ contém aqueles elementos que estão em todos os três conjuntos *A, B e C*. Então,

$$A \cap B \cap C = \{0\}.$$ ◀

Podemos também considerar uniões e interseções de um número arbitrário de conjuntos. Usamos as definições abaixo.

DEFINIÇÃO 6 A *união* de uma coleção de conjuntos é o conjunto que contém aqueles elementos que são membros de, pelo menos, um dos conjuntos da coleção.

Usamos a notação

$$A_1 \cup A_2 \cup \cdots \cup A_n = \bigcup_{i=1}^{n} A_i$$

para indicar a união dos conjuntos $A_1, A_2, \ldots, A_n$.

DEFINIÇÃO 7 A *interseção* de uma coleção de conjuntos é o conjunto que contém aqueles elementos que são membros de todos os conjuntos da coleção.

Usamos a notação

$$A_1 \cap A_2 \cap \cdots \cap A_n = \bigcap_{i=1}^{n} A_i$$

para indicar a interseção dos conjuntos $A_1, A_2, \ldots, A_n$. Ilustramos as uniões e interseções generalizadas com o Exemplo 16.

EXEMPLO 16 Considere $A_i = \{i, i+1, i+2, \ldots\}$. Então,

$$\bigcup_{i=1}^{n} A_i = \bigcup_{i=1}^{n} \{i, i+1, i+2, \ldots\} = \{1, 2, 3, \ldots\},$$

e

$$\bigcap_{i=1}^{n} A_i = \bigcap_{i=1}^{n} \{i, i+1, i+2, \ldots\} = \{n, n+1, n+2, \ldots\}.$$ ◀

Podemos estender a notação que introduzimos para uniões e interseções a outras famílias de conjuntos. Em particular, podemos usar a notação

$$A_1 \cup A_2 \cup \cdots \cup A_n \cup \cdots = \bigcup_{i=1}^{\infty} A_i$$

para indicar a união dos conjuntos $A_1, A_2, \ldots, A_n, \ldots$. Semelhantemente, a interseção desses conjuntos pode ser indicada por

$$A_1 \cap A_2 \cap \cdots \cap A_n \cap \cdots = \bigcap_{i=1}^{\infty} A_i.$$

Mais usualmente, as notações $\bigcap_{i \in I} A_i$ e $\bigcup_{i \in I} A_i$ são usadas para indicar a interseção e união dos conjuntos A_i para $i \in I$, respectivamente. Note que temos $\bigcap_{i \in I} A_i = \{x \mid \forall i \in I (x \in A_i)\}$ e $\bigcup_{i \in I} A_i = \{x \mid \exists i \in I (x \in A_i)\}$.

EXEMPLO 17 Suponha que $A_i = \{1, 2, 3, \ldots, i\}$ para $i = 1, 2, 3, \ldots$. Então,

$$\bigcup_{i=1}^{\infty} A_i = \bigcup_{i=1}^{\infty} \{1, 2, 3, \ldots, i\} = \{1, 2, 3, \ldots\} = \mathbf{Z}^+$$

e

$$\bigcap_{i=1}^{\infty} A_i = \bigcap_{i=1}^{\infty} \{1, 2, 3, \ldots, i\} = \{1\}.$$

Ao ver que a união desses conjuntos é o conjunto dos números inteiros positivos, note que todo número inteiro positivo está, em pelo menos, um desses conjuntos, porque o inteiro n pertence a $A_n = \{1, 2, \ldots, n\}$ e todo elemento dos conjuntos na união é um número inteiro positivo. Ao ver a interseção desses conjuntos, note que o único elemento que pertence a todos os conjuntos $A_1, A_2, \ldots$ é 1. Para ver isso, note que $A_1 = \{1\}$ e $1 \in A_i$ para $i = 1, 2, \ldots$. ◀

Representação Computacional de Conjuntos

Há várias maneiras de representar conjuntos usando um computador. Um método é juntar os elementos do conjunto de forma não ordenada. Entretanto, se isso for feito, as operações para computar a união, intersecção ou diferença de dois conjuntos podem demorar muito, pois cada uma dessas operações exigiria uma grande busca pelos elementos. Apresentaremos um método para armazenar elementos usando uma ordem arbitrária dos elementos do conjunto universo. Esse método de representar conjuntos torna mais fácil a combinação de conjuntos em computadores.

Suponha que o conjunto universo U seja finito (e de tamanho razoável para que o número de elementos de U não seja maior que a memória do computador a ser usado). Primeiro, especifique uma ordem arbitrária dos elementos de U, por exemplo, $a_1, a_2, \ldots, a_n$. Represente um subconjunto A de U com a cadeia de bits de extensão n, em que o i-ésimo bit nessa cadeia é 1, se a_i pertencer a A, e 0, se a_i não pertencer a A. O Exemplo 18 irá mostrar essa técnica.

EXEMPLO 18 Considere $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, em que a ordem dos elementos de U é crescente, ou seja, $a_i = i$. Quais cadeias de bits representam o subconjunto de todos os números inteiros ímpares em U, o subconjunto de todos os números inteiros pares em U e o subconjunto dos números inteiros não excedentes a 5 em U?

Solução: A cadeia de bits que representa o conjunto dos números inteiros ímpares em U, ou seja, $\{1, 3, 5, 7, 9\}$, tem um bit 1 na primeira, terceira, quinta, sétima e nona posições e um zero nas demais posições. Ou seja, é

10 1010 1010.

(Separamos esta cadeia de bits de extensão dez por sua dificuldade de leitura.) Semelhantemente, representamos o subconjunto de todos os números inteiros pares em U, ou seja, $\{2, 4, 6, 8, 10\}$, pela cadeia

01 0101 0101.

O conjunto de todos os números inteiros em U que não excedem 5, ou seja, $\{1, 2, 3, 4, 5\}$, é representado pela cadeia

11 1110 0000. ◀

Usar cadeia de bits para representar conjuntos é uma maneira que facilita o trabalho quando temos de encontrar os complementos, uniões, intersecções e diferenças de conjuntos. Para encontrar uma cadeia de bits para o complemento de um conjunto a partir da cadeia de bits desse conjunto, nós simplesmente trocamos todo 1 por 0 e todo 0 por 1, porque $x \in A$ se e somente se $x \notin \overline{A}$. Note que esta operação corresponde a assumir a negação de cada bit quando associamos um bit com um valor-verdade — com 1 representando verdadeiro e 0, falso.

EXEMPLO 19 Vimos que a cadeia de bits para o conjunto $\{1, 3, 5, 7, 9\}$ (com o conjunto universo $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$) é

10 1010 1010.

Qual é a cadeia de bits para o complemento desse conjunto?

Solução: A cadeia de bits para o complemento desse conjunto é obtida pela troca dos 0s por 1s e vice-versa. A partir disto, temos a cadeia

01 0101 0101,

que corresponde ao conjunto $\{2, 4, 6, 8, 10\}$. ◀

Para obter a cadeia de bits para a união e intersecção de dois conjuntos, aplicamos as operações booleanas de lógica binária nas cadeias de bits que representam os dois conjuntos. O bit na i-ésima posição da união é 1 se um dos bits na i-ésima posição nas duas cadeias for 1 (ou ambos) e é 0 quando ambos os bits forem 0. Assim, a cadeia de bits para a união é o operador *OR* da cadeia de bits para os dois conjuntos. O bit na i-ésima posição da cadeia de bits da intersecção é 1 quando os bits na posição correspondente nas duas cadeias forem 1, e 0, quando um dos dois bits for 0 (ou ambos). Assim, a cadeia de bits para a intersecção é o operador *AND* das cadeias de bits para esses dois conjuntos.

EXEMPLO 20 As cadeias de bits para os conjuntos $\{1, 2, 3, 4, 5\}$ e $\{1, 3, 5, 7, 9\}$ são 11 1110 0000 e 10 1010 1010, respectivamente. Use as cadeias de bits para encontrar a união e a intersecção desses conjuntos.

Solução: A cadeia de bits para a união desses conjuntos é

$$11\ 1110\ 0000 \vee 10\ 1010\ 1010 = 11\ 1110\ 1010,$$

que corresponde ao conjunto $\{1, 2, 3, 4, 5, 7, 9\}$. A cadeia de bits para a intersecção desses conjuntos é

$$11\ 1110\ 0000 \wedge 10\ 1010\ 1010 = 10\ 1010\ 0000,$$

que corresponde ao conjunto $\{1, 3, 5\}$. ◀

Exercícios

1. Seja A o conjunto de estudantes que mora a um quilômetro de distância da faculdade e B, o conjunto dos estudantes que vão a pé para as aulas. Descreva quais são os estudantes em cada um dos conjuntos abaixo.
a) $A \cap B$ **b)** $A \cup B$
c) $A - B$ **d)** $B - A$

2. Suponha que A seja o conjunto dos veteranos de sua faculdade e B, o conjunto dos estudantes de matemática discreta de sua faculdade. Expresse cada um dos conjuntos abaixo em termos de A e B.
a) o conjunto dos veteranos que assistem às aulas de matemática discreta em sua faculdade
b) o conjunto dos veteranos que não assistem às aulas de matemática discreta em sua faculdade
c) o conjunto dos estudantes que são ou veteranos ou assistem às aulas de matemática discreta em sua faculdade
d) o conjunto dos estudantes que ou não são veteranos ou não assistem às aulas de matemática discreta em sua faculdade

3. Considere $A = \{1, 2, 3, 4, 5\}$ e $B = \{0, 3, 6\}$. Encontre
a) $A \cup B$. **b)** $A \cap B$.
c) $A - B$. **d)** $B - A$.

4. Considere $A = \{a, b, c, d, e\}$ e $B = \{a, b, c, d, e, f, g, h\}$. Encontre
a) $A \cup B$.
b) $A \cap B$.
c) $A - B$.
d) $B - A$.

Nos exercícios 5 a 10, suponha que A seja um subconjunto de algum conjunto universo U.

5. Demonstre a propriedade da complementação da Tabela 1 mostrando que $\overline{\overline{A}} = A$.

6. Demonstre as propriedades dos elementos neutros e da Tabela 1 mostrando que
a) $A \cup \emptyset = A$. **b)** $A \cap U = A$.

7. Demonstre as propriedades de dominação da Tabela 1 mostrando que
a) $A \cup U = U$.
b) $A \cap \emptyset = \emptyset$.

8. Demonstre as propriedades idempotentes da Tabela 1 mostrando que
a) $A \cup A = A$.
b) $A \cap A = A$.

9. Demonstre as propriedades dos complementares da Tabela 1 mostrando que
a) $A \cup \overline{A} = U$.
b) $A \cap \overline{A} = \emptyset$.

10. Mostre que
a) $A - \emptyset = A$.
b) $\emptyset - A = \emptyset$.

11. Considere A e B como conjuntos. Demonstre as propriedades comutativas da Tabela 1 mostrando que
a) $A \cup B = B \cup A$.
b) $A \cap B = B \cap A$.

12. Demonstre a primeira propriedade de absorção da Tabela 1 mostrando que se A e B forem conjuntos, então $A \cup (A \cap B) = A$.

13. Demonstre a segunda propriedade de absorção da Tabela 1 mostrando que se A e B forem conjuntos, então $A \cap (A \cup B) = A$.

14. Encontre os conjuntos A e B, se $A - B = \{1, 5, 7, 8\}$, $B - A = \{2, 10\}$ e $A \cap B = \{3, 6, 9\}$.

15. Demonstre a primeira lei de De Morgan da Tabela 1 mostrando que se A e B forem conjuntos, então $\overline{A \cup B} = \overline{A} \cap \overline{B}$

a) mostrando que cada lado é um subconjunto do outro lado.
b) usando uma tabela de pertinência.

16. Considere A e B como conjuntos. Mostre que
a) $(A \cap B) \subseteq A$.
b) $A \subseteq (A \cup B)$.
c) $A - B \subseteq A$.
d) $A \cap (B - A) = \emptyset$.
e) $A \cup (B - A) = A \cup B$.

17. Mostre que se A, B e C são conjuntos, então $\overline{A \cap B \cap C} = \overline{A} \cup \overline{B} \cup \overline{C}$.

a) mostrando que cada lado é um subconjunto do outro lado.
b) usando uma tabela de pertinência.

18. Considere A, B e C como conjuntos. Mostre que
a) $(A \cup B) \subseteq (A \cup B \cup C)$.
b) $(A \cap B \cap C) \subseteq (A \cap B)$.
c) $(A - B) - C \subseteq A - C$.
d) $(A - C) \cap (C - B) = \emptyset$.
e) $(B - A) \cup (C - A) = (B \cup C) - A$.

19. Mostre que se A e B são conjuntos, então $A - B = A \cap \overline{B}$.

20. Mostre que se A e B são conjuntos, então $(A \cap B) \cup (A \cap \overline{B}) = A$.

21. Demonstre a primeira propriedade associativa da Tabela 1, mostrando que se A, B e C são conjuntos, então $A \cup (B \cup C) = (A \cup B) \cup C$.

22. Demonstre a segunda propriedade associativa da Tabela 1, mostrando que se A, B e C são conjuntos, então $A \cap (B \cap C) = (A \cap B) \cap C$.

23. Demonstre a segunda propriedade distributiva da Tabela 1, mostrando que se A, B e C são conjuntos, então $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.

24. Considere A, B e C como conjuntos. Mostre que $(A - B) - C = (A - C) - (B - C)$.

25. Considere $A = \{0, 2, 4, 6, 8, 10\}$, $B = \{0, 1, 2, 3, 4, 5, 6\}$ e $C = \{4, 5, 6, 7, 8, 9, 10\}$. Encontre
a) $A \cap B \cap C$. **b)** $A \cup B \cup C$.
c) $(A \cup B) \cap C$. **d)** $(A \cap B) \cup C$.

26. Desenhe o diagrama de Venn para cada uma das combinações abaixo dos conjuntos A, B e C.
a) $A \cap (B \cup C)$ **b)** $\overline{A} \cap \overline{B} \cap \overline{C}$
c) $(A - B) \cup (A - C) \cup (B - C)$

27. Desenhe o diagrama de Venn para cada uma das combinações abaixo dos conjuntos A, B e C.
a) $A \cap (B - C)$ **b)** $(A \cap B) \cup (A \cap C)$
c) $(A \cap \overline{B}) \cup (A \cap \overline{C})$

28. Desenhe o diagrama de Venn para cada uma das combinações abaixo dos conjuntos A, B, C e D.
a) $(A \cap B) \cup (C \cap D)$ **b)** $\overline{A} \cup \overline{B} \cup \overline{C} \cup \overline{D}$
c) $A - (B \cap C \cap D)$

29. O que você pode dizer sobre os conjuntos A e B se sabemos que
a) $A \cup B = A$? **b)** $A \cap B = A$?
c) $A - B = A$? **d)** $A \cap B = B \cap A$?
e) $A - B = B - A$?

30. Você pode concluir que $A = B$ se A, B e C são conjuntos, tal que
a) $A \cup C = B \cup C$? **b)** $A \cap C = B \cap C$?
c) $A \cup B = B \cup C$ e $A \cap C = B \cap C$?

31. Considere A e B como subconjuntos de um conjunto universo U. Mostre que $A \subseteq B$ se e somente se $\overline{B} \subseteq \overline{A}$.

A **diferença simétrica** de A e B, indicada por $A \oplus B$, é o conjunto que contém aqueles elementos que estão em A ou B, mas não em ambos.

32. Encontre a diferença simétrica de $\{1, 3, 5\}$ e $\{1, 2, 3\}$.

33. Encontre a diferença simétrica do conjunto de estudantes de ciência da computação e do conjunto de estudantes de matemática de sua faculdade.

34. Desenhe um diagrama de Venn para a diferença simétrica dos conjuntos A e B.

35. Mostre que $A \oplus B = (A \cup B) - (A \cap B)$.

36. Mostre que $A \oplus B = (A - B) \cup (B - A)$.

37. Mostre que A é um subconjunto de um conjunto universo U, então
a) $A \oplus A = \emptyset$. **b)** $A \oplus \emptyset = A$.
c) $A \oplus U = \overline{A}$. **d)** $A \oplus \overline{A} = U$.

38. Mostre que se A e B são conjuntos, então
a) $A \oplus B = B \oplus A$. **b)** $(A \oplus B) \oplus B = A$.

39. O que você pode dizer sobre os conjuntos A e B, se $A \oplus B = A$?

***40.** Determine se a diferença simétrica é associativa, ou seja, se A, B e C são conjuntos, temos que $A \oplus (B \oplus C) = (A \oplus B) \oplus C$?

***41.** Suponha que A, B e C sejam conjuntos, tal que $A \oplus C = B \oplus C$. É o caso de $A = B$?

42. Se A, B, C e D são conjuntos, temos que $(A \oplus B) \oplus (C \oplus D) = (A \oplus C) \oplus (B \oplus D)$?

43. Se A, B, C e D são conjuntos, temos que $(A \oplus B) \oplus (C \oplus D) = (A \oplus D) \oplus (B \oplus C)$?

***44.** Mostre que se A, B e C são conjuntos, então

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|.$$

(Este é um caso especial do princípio da inclusão-exclusão, que será estudado no Capítulo 7.)

***45.** Considere $A_i = \{1, 2, 3, \ldots, i\}$ para $i = 1, 2, 3, \ldots$. Encontre
a) $\bigcup_{i=1}^{n} A_i$. **b)** $\bigcap_{i=1}^{n} A_i$.

***46.** Considere $A_i = \{\ldots, -2, -1, 0, 1, \ldots, i\}$. Encontre
a) $\bigcup_{i=1}^{n} A_i$. **b)** $\bigcap_{i=1}^{n} A_i$.

47. Considere A_i como o conjunto de todas as cadeias de bits não vazias (ou seja, cadeias de bits com, pelo menos, um de extensão) de extensão não excedente a i. Encontre
a) $\bigcup_{i=1}^{n} A_i$. **b)** $\bigcap_{i=1}^{n} A_i$.

48. Encontre $\bigcup_{i=1}^{\infty} A_i$ e $\bigcap_{i=1}^{\infty} A_i$ se para todo número inteiro positivo i,
- **a)** $A_i = \{i, i+1, i+2, \ldots\}$.
- **b)** $A_i = \{0, i\}$.
- **c)** $A_i = (0, i)$, ou seja, o conjunto dos números reais x com $0 < x < i$.
- **d)** $A_i = (i, \infty)$, ou seja, o conjunto dos números reais x com $x > i$.

49. Encontre $\bigcup_{i=1}^{\infty} A_i$ e $\bigcap_{i=1}^{\infty} A_i$ se para todo número inteiro positivo i,
- **a)** $A_i = \{-i, -i+1, \ldots, -1, 0, 1, \ldots, i-1, i\}$.
- **b)** $A_i = \{-i, i\}$.
- **c)** $A_i = [-i, i]$, ou seja, o conjunto dos números reais x com $-i \le x \le i$.
- **d)** $A_i = [i, \infty]$, ou seja, o conjunto dos números reais x com $x \ge i$.

50. Suponha que o conjunto universo seja $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Expresse cada um dos conjuntos abaixo com cadeias de bits, em que o i-ésimo bit na cadeia é 1, se i estiver no conjunto, e 0, se não estiver no conjunto.
- **a)** $\{3, 4, 5\}$
- **b)** $\{1, 3, 6, 10\}$
- **c)** $\{2, 3, 4, 7, 8, 9\}$

51. Use o mesmo conjunto universo do exercício anterior para encontrar o conjunto específico para cada uma das cadeias abaixo.
- **a)** 11 1100 1111
- **b)** 01 0111 1000
- **c)** 10 0000 0001

52. Quais os subconjuntos de um conjunto universo finito que são representados pelas cadeias de bits abaixo?
- **a)** a cadeia apenas com zeros
- **b)** a cadeia apenas com uns

53. Qual é a cadeia de bits, correspondente à diferença de dois conjuntos?

54. Qual é a cadeia de bits, correspondente à diferença simétrica de dois conjuntos?

55. Mostre como os operadores binários nas cadeias de bits podem ser usados para encontrar as combinações de $A = \{a, b, c, d, e\}$, $B = \{b, c, d, g, p, t, v\}$, $C = \{c, e, i, o, u, x, y, z\}$ e $D = \{d, e, h, i, n, o, t, u, x, y\}$.
- **a)** $A \cup B$
- **b)** $A \cap B$
- **c)** $(A \cup D) \cap (B \cup C)$
- **d)** $A \cup B \cup C \cup D$

56. Como a união e a intersecção de n conjuntos, todos subconjuntos do conjunto universo U, podem ser encontradas usando cadeias de bits?

O **sucessor** do conjunto A é o conjunto $A \cup \{A\}$.

57. Encontre os sucessores dos conjuntos abaixo.
- **a)** $\{1, 2, 3\}$
- **b)** $\emptyset$
- **c)** $\{\emptyset\}$
- **d)** $\{\emptyset, \{\emptyset\}\}$

58. Quantos elementos o sucessor de um conjunto com n elementos tem?

Às vezes o número de vezes que um elemento ocorre em uma coleção não ordenada é relevante. **Multiconjuntos** são coleções não ordenadas de elementos, em que cada elemento pode ocorrer mais de uma vez. A notação $\{m_1 \cdot a_1 \cdot m_2 \cdot a_2, \ldots m_r \cdot a_r\}$ indica o multiconjunto com o elemento a_1 aparecendo m_1 vezes, o elemento a_2 aparecendo m_2 vezes e assim por diante. Os números m_i, $i = 1, 2, \ldots, r$ são chamados de **multiplicidades** dos elementos a_i, $i = 1, 2, \ldots, r$.

Considere P e Q como dois multiconjuntos. A **união** dos multiconjuntos P e Q é o multiconjunto em que a multiplicidade de um elemento é o máximo de suas multiplicidades em P e Q. A **interseção** de P e Q é o multiconjunto em que a multiplicidade de um elemento é o mínimo das multiplicidades em P e Q. A **diferença** de P e Q é o multiconjunto em que a multiplicidade de um elemento é a multiplicidade do elemento em P menos a multiplicidade em Q, a menos que sua diferença seja negativa, em que a multiplicidade será 0. A **soma** de P e Q é o multiconjunto em que a multiplicidade de um elemento é a soma das multiplicidades em P e Q. A união, intersecção e diferença de P e Q são indicadas por $P \cup Q$, $P \cap Q$, e $P - Q$, respectivamente (em que essas operações não deverão ser confundidas com suas análogas operações para conjuntos). A soma de P e Q é indicada por $P + Q$.

59. Considere A e B como os multiconjuntos $\{3 \cdot a, 2 \cdot b, 1 \cdot c\}$ e $\{2 \cdot a, 3 \cdot b, 4 \cdot d\}$, respectivamente. Encontre
- **a)** $A \cup B$.
- **b)** $A \cap B$.
- **c)** $A - B$.
- **d)** $B - A$.
- **e)** $A + B$.

60. Suponha que A seja o multiconjunto que tenha como seus elementos os tipos de equipamentos de computadores necessários para um departamento da universidade, em que as multiplicidades são os números de partes que cada tipo necessita e B é o multiconjunto análogo para um segundo departamento da universidade. Por exemplo, A poderia ser o multiconjunto $\{107 \cdot$ computadores, $44 \cdot$ routers, $6 \cdot$ servidores$\}$ e B poderia ser o multiconjunto $\{14 \cdot$ computadores, $6 \cdot$ routers, $2 \cdot$ computadores principais$\}$.
- **a)** Qual combinação de A e B representa o equipamento que a universidade deveria comprar, assumindo que ambos os departamentos usam o mesmo equipamento?
- **b)** Qual combinação de A e B representa o equipamento que será usado por ambos os departamentos se ambos usam o mesmo equipamento?
- **c)** Qual combinação de A e B representa o equipamento que o segundo departamento usa, mas não o primeiro departamento, se ambos os departamentos usam o mesmo equipamento?
- **d)** Qual combinação de A e B representa o equipamento que a universidade deverá comprar se os departamentos não dividem equipamento?

Conjuntos fuzzy são usados em inteligência artificial. Cada elemento no conjunto universo U tem um **grau de pertinência** em um conjunto fuzzy S, o qual é um número real entre 0 e 1 (incluindo 0 e 1). O conjunto fuzzy S é indicado pela lista de elementos com seus graus de pertinência (elementos com grau 0 de pertinência não são listados). Por exemplo, escrevemos {0,6 Alice, 0,9 Brian, 0,4 Fred, 0,1 Oscar, 0,5 Rita} para o conjunto F (de pessoas famosas) para indicar que Alice tem um grau 0,6 de pertinência em F, Brian tem um grau 0,9 de pertinência em F, Fred tem um grau 0,4 de pertinência em F, Oscar tem um grau 0,1 de pertinência em F e Rita tem um grau 0,5 de pertinência

em F (assim, Brian é o mais famoso e Oscar o menos famoso dessas pessoas). Suponha também que R seja o conjunto de pessoas ricas com $R = \{0{,}4$ Alice, 0,8 Brian, 0,2 Fred, 0,9 Oscar, 0,7 Rita$\}$.

61. O **complemento** de um conjunto fuzzy S é o conjunto $\overline{S}$, com o grau de pertinência de um elemento em $\overline{S}$ igual a 1 menos o grau de pertinência desse elemento em S. Encontre $\overline{F}$ (o conjunto fuzzy das pessoas que não são famosas) e $\overline{R}$ (o conjunto fuzzy das pessoas que não são ricas).

62. A **união** de dois conjuntos fuzzy S e T é o conjunto fuzzy $S \cup T$, em que o grau de pertinência de um elemento em $S \cup T$ é o máximo dos graus de pertinência desse elemento em S e em T. Encontre o conjunto fuzzy $F \cup R$ das pessoas ricas ou famosas.

63. A **interseção** de dois conjuntos fuzzy S e T é o conjunto fuzzy $S \cap T$, em que o grau de pertinência de um elemento em $S \cap T$ é o mínimo dos graus de pertinência desse elemento em S e em T. Encontre o conjunto fuzzy $F \cap R$ de pessoas ricas e famosas.

2.3 Funções

Introdução

Em muitos momentos delimitamos, para cada elemento de um conjunto, um determinado elemento de um segundo conjunto (o qual pode ser o mesmo do primeiro). Por exemplo, suponha que para cada estudante na aula de matemática discreta é determinada uma nota que é uma letra do conjunto $\{A, B, C, D, F\}$. Suponha que a nota de cada aluno é: A para Adams, C para Chou, B para Goodfriend, A para Rodriguez e F para Stevens. Essa determinação das notas é ilustrada na Figura 1.

Essa determinação é um exemplo de função. O conceito de função é extremamente importante em matemática e em ciência da computação. Por exemplo, em matemática discreta as funções são usadas na definição de estruturas discretas, como seqüências e cadeias. As funções são também usadas para representar quanto tempo um computador leva para resolver problemas de um determinado tamanho. Muitos programas de computação e sub-rotinas são criados para calcular valores de funções. As funções recursivas, que são funções definidas em termos delas mesmas, são usadas pela ciência da computação; elas serão estudadas no Capítulo 4. Esta seção revê os conceitos básicos que envolvem funções necessárias para a matemática discreta.

DEFINIÇÃO 1 Sejam A e B conjuntos não vazios. Uma *função* f de A para B é uma determinação de exatamente um elemento de B para cada elemento de A. Escrevemos $f(a) = b$ se b for o único elemento de B determinado pela função f para o elemento a de A. Se f é uma função de A para B, escrevemos $f : A \rightarrow B$.

Lembre-se: As funções são também chamadas de **mapeamentos** ou **transformações.**

Auto-avaliação

As funções são específicas em muitos casos. Às vezes, constatamos explicitamente as determinações, como na Figura 1. Normalmente damos uma fórmula, como $f(x) = x + 1$, para definir uma função. Outras vezes, usamos um programa de computação para especificar uma função.

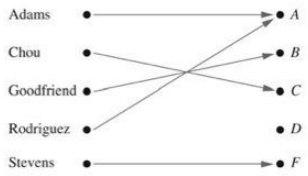


FIGURA 1 Determinação de Notas em uma Sala de Matemática Discreta.

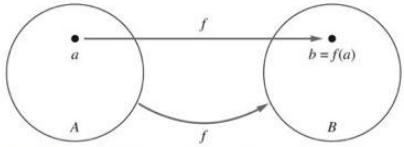


FIGURA 2 A Função *f* Mapeia *A* para *B*.

Uma função $f : A \to B$ pode também ser definida em termos de uma relação de A para B. Recorde a Seção 2.1, em que uma relação de A para B é apenas um subconjunto de $A \times B$. Uma relação de A para B que contém um, e apenas um, par ordenado (a, b) para cada elemento $a \in A$, define uma função f de A para B. Essa função é definida pela determinação $f(a) = b$, em que (a, b) é o único par ordenado na relação que tem a como seu primeiro elemento.

DEFINIÇÃO 2 Se f é uma função de A para B, dizemos que A é o *domínio* de f e B é o *contradomínio* de f. Se $f(a) = b$, dizemos que b é a *imagem* de a e a é a *imagem inversa* de b. A *imagem* de f é o conjunto com todas as imagens dos elementos de A. Além disso, se f é uma função de A para B, dizemos que f *mapeia* A para B.

A Figura 2 representa uma função f de A para B.

Quando definimos uma função, especificamos seu domínio, seu contradomínio e o mapeamento dos elementos no domínio e no contradomínio. Duas funções são **iguais** quando elas apresentam o mesmo domínio, têm o mesmo contradomínio e mapeiam os elementos de seus domínios comuns para os mesmos elementos no seu contradomínio. Note que se trocarmos o domínio ou o contradomínio de uma função, então obteremos uma função diferente. Se trocarmos o mapeamento dos elementos, também obteremos uma função diferente.

Os exemplos de 1 a 5 fornecem exemplos de funções. Em cada caso, descrevemos o domínio, o contradomínio, o conjunto imagem e as determinações de valores para os elementos do domínio.

EXEMPLO 1 Qual é o domínio, o contradomínio e o conjunto imagem da função que determina as notas dos estudantes descritos no primeiro parágrafo da introdução desta seção?

Solução: Considere G como a função que determina uma nota para um estudante na sala de matemática discreta. Note que $G(\text{Adams}) = A$, por exemplo. O domínio de G é o conjunto {Adams, Chou, Goodfriend, Rodriguez, Stevens} e o contradomínio é o conjunto $\{A, B, C, D, F\}$. O conjunto imagem de G é o conjunto $\{A, B, C, F\}$, pois cada nota, exceto D, é designada a algum estudante. ◀

EXEMPLO 2 Considere R como a relação dos pares ordenados (Abdul, 22), (Brenda, 24), (Carla, 21), (Desire, 22), (Eddie, 24) e (Felicia, 22), em que cada par é formado por um estudante e por sua idade. Qual é a função que essa relação determina?

Solução: Essa relação define a função f, com $f(\text{Abdul}) = 22$, $f(\text{Brenda}) = 24$, $f(\text{Carla}) = 21$, $f(\text{Desire}) = 22$, $f(\text{Eddie}) = 24$ e $f(\text{Felicia}) = 22$. Aqui, o domínio é o conjunto {Abdul, Brenda, Carla, Desire, Eddie, Felicia}. Para definir a função f, precisamos especificar um contradomínio. Aqui, podemos entender o contradomínio como sendo o conjunto de números inteiros positivos

que garantem que o contradomínio tenha todas as possíveis idades dos estudantes. (Note que poderíamos escolher um contradomínio menor, mas isso mudaria a função.) Por fim, o conjunto imagem é o conjunto {21, 22, 24}. ◀

EXEMPLO 3 Considere f como a função que determina os dois últimos bits de uma cadeia de bits de extensão 2, ou maior. Por exemplo, $f(11010) = 10$. Então, o domínio de f é o conjunto de todas as cadeias de bits de extensão 2, ou mais, e o contradomínio e o conjunto imagem são o conjunto {00, 01, 10, 11}. ◀

Exemplos Extras

EXEMPLO 4 Considere $f: \mathbf{Z} \to \mathbf{Z}$ como a função que determina o quadrado de um número inteiro para este inteiro. Então, $f(x) = x^2$, em que o domínio de f é o conjunto de todos os inteiros, o contradomínio de f, todos os números inteiros, e o conjunto imagem de f, o conjunto de todos os inteiros que são quadrados perfeitos, ou seja, {0, 1, 4, 9, . . . }. ◀

EXEMPLO 5 O domínio e o contradomínio de funções são geralmente especificados em linguagem computacional. Por exemplo, o enunciado em Java

```
int floor(float real) {. . .}
```

e o enunciado de Pascal

```
function floor(x: real): integer
```

ambos afirmam que o domínio dessa função é o conjunto dos números reais, e seu contradomínio, o conjunto dos números inteiros. ◀

Duas funções com valores reais com o mesmo domínio podem ser somadas e multiplicadas.

DEFINIÇÃO 3 Sejam f_1 e f_2 funções de A para **R**. Então, $f_1 + f_2$ e $f_1 f_2$ também são funções de A para **R** definidas por

$$(f_1 + f_2)(x) = f_1(x) + f_2(x),$$
$$(f_1 f_2)(x) = f_1(x) f_2(x).$$

Note que as funções $f_1 + f_2$ e $f_1 f_2$ foram definidas especificando seus valores em x em termos de valores de f_1 e f_2 em x.

EXEMPLO 6 Considere f_1 e f_2 como funções de **R** a **R**, tal que $f_1(x) = x^2$ e $f_2(x) = x - x^2$. Quais são as funções $f_1 + f_2$ e $f_1 f_2$?

Solução: A partir da definição de soma e produto de funções, temos que

$$(f_1 + f_2)(x) = f_1(x) + f_2(x) = x^2 + (x - x^2) = x$$

e

$$(f_1 f_2)(x) = x^2(x - x^2) = x^3 - x^4.$$ ◀

Quando f é uma função de um conjunto A para um conjunto B, a imagem de um subconjunto de A também pode ser definida.

DEFINIÇÃO 4 Seja f uma função do conjunto A para o conjunto B e seja S um subconjunto de A. A *imagem* de S sob a função f é o subconjunto de B que é formado pelas imagens dos elementos de S. Indicamos a imagem de S por $f(S)$, então

$$f(S) = \{t \mid \exists s \in S\,(t = f(s))\}.$$

Usamos também a notação abreviada $\{f(s) \mid s \in S\}$ para indicar tal conjunto.

Lembre-se: A notação $f(S)$ para a imagem do conjunto S sob a função f é potencialmente ambígua. Aqui, $f(S)$ indica um conjunto e não o valor da função f para o conjunto S.

EXEMPLO 7 Considere $A = \{a, b, c, d, e\}$ e $B = \{1, 2, 3, 4\}$ com $f(a) = 2$, $f(b) = 1$, $f(c) = 4$, $f(d) = 1$ e $f(e) = 1$. A imagem do subconjunto $S = \{b, c, d\}$ é o conjunto $f(S) = \{1, 4\}$. ◀

Funções Injetoras e Sobrejetoras

Algumas funções nunca determinam o mesmo valor para dois elementos diferentes no domínio. Essas funções são chamadas de **injetoras.**

DEFINIÇÃO 5 Uma função f é chamada de *injetora,* ou *um para um,* se e somente se $f(a) = f(b)$ implica que $a = b$ para todos os a e b no domínio de f. Uma função pode ser chamada de *injeção* se for de um para um.

Note que uma função f é injetora se e somente se $f(a) \neq f(b)$ sempre que $a \neq b$. Essa maneira de expressar que f é injetora é obtida a partir da contraposição da implicação na definição.

Lembre-se: Podemos expressar que f é injetora usando quantificadores como $\forall a\, \forall b\, (f(a) = f(b) \rightarrow a = b)$ ou, equivalentemente, $\forall a\, \forall b\, (a \neq b \rightarrow f(a) \neq f(b))$, em que o universo do discurso é o domínio da função.

Auto-avaliação Ilustramos este conceito dando alguns exemplos de funções que são injetoras e outras funções que não são injetoras.

EXEMPLO 8 Determine se a função f de $\{a, b, c, d\}$ para $\{1, 2, 3, 4, 5\}$ com $f(a) = 4$, $f(b) = 5$, $f(c) = 1$ e $f(d) = 3$ é injetora.

Exemplos Extras *Solução:* A função f é injetora, pois f assume diferentes valores para os quatro elementos no seu domínio. Isso é ilustrado na Figura 3. ◀

EXEMPLO 9 Determine se a função $f(x) = x^2$ do conjunto de número inteiros para o conjunto de números inteiros é injetora.

Solução: A função $f(x) = x^2$ não é injetora, porque, por exemplo, $f(1) = f(-1) = 1$, mas $1 \neq -1$.

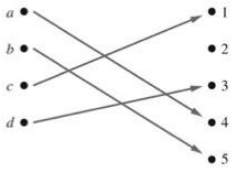


FIGURA 3 Uma Função Injetora.

Note que a função $f(x) = x^2$ com o domínio restrito a $\mathbf{Z}^+$ é injetora. (Tecnicamente, quando restringimos o domínio de uma função, obtemos uma nova função cujos valores concordam com aqueles da função original para os elementos do domínio restrito. A função restrita não é definida pelos elementos do domínio original, mas sim do domínio restrito.) ◀

EXEMPLO 10 Determine se a função $f(x) = x + 1$ do conjunto de números reais para ele mesmo é injetora.

Solução: A função $f(x) = x + 1$ é uma função injetora. Para demonstrar isso, note que $x + 1 \neq y + 1$ quando $x \neq y$. ◀

Daremos agora algumas condições para garantir que uma função seja injetora.

DEFINIÇÃO 6 Uma função f cujo domínio e contradomínio são subconjuntos do conjunto dos números reais é chamada de *crescente* se $f(x) \leq f(y)$, e *estritamente crescente* se $f(x) < f(y)$, sempre que $x < y$ e x e y estão no domínio de f. Analogamente, f é chamada de *decrescente* se $f(x) \geq f(y)$, e *estritamente decrescente* se $f(x) > f(y)$, sempre que $x < y$ e x e y estão no domínio de f. (A palavra *estritamente* nesta definição indica uma inequação estrita.)

Lembre-se: Uma função f é crescente se $\forall x\, \forall y\, (x < y \rightarrow f(x) \leq f(y))$, estritamente crescente se $\forall x\, \forall y\, (x < y \rightarrow f(x) < f(y))$, decrescente se $\forall x\, \forall y\, (x < y \rightarrow f(x) \geq f(y))$ e estritamente decrescente se $\forall x\, \forall y\, (x < y \rightarrow f(x) > f(y))$, em que o universo de discurso é o domínio de f.

A partir dessas definições, vemos que uma função que é ou estritamente crescente ou estritamente decrescente deve ser injetora. Entretanto, uma função que é crescente, mas não estritamente crescente, ou decrescente, mas não estritamente decrescente, não é necessariamente injetora.

Para algumas funções, o conjunto imagem e o contradomínio são iguais. Ou seja, todo membro do contradomínio é a imagem de algum elemento do domínio. Funções com esta propriedade são chamadas de funções **sobrejetoras** ou **sobrejetivas**.

DEFINIÇÃO 7 Uma função f de A para B é chamada de *sobrejetora* ou *sobrejetiva,* se e somente se para todo elemento $b \in B$ houver um elemento $a \in A$ com $f(a) = b$.

Lembre-se: Uma função f é sobrejetiva se $\forall y\, \exists x\, (f(x) = y)$, em que o domínio para x é o domínio da função e o domínio para y é o contradomínio da função.

Daremos agora alguns exemplos de funções sobrejetivas e funções que não são sobrejetivas.

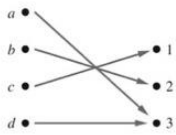


FIGURA 4 Uma Função Sobrejetiva.

EXEMPLO 11 Considere f como a função de $\{a, b, c, d\}$ para $\{1, 2, 3\}$ definida por $f(a) = 3$, $f(b) = 2$, $f(c) = 1$, e $f(d) = 3$. A função f é uma função sobrejetiva?

Exemplos Extras

Solução: Pelo fato de todos os três elementos do contradomínio serem imagens de elementos no domínio, vemos que f é sobrejetiva. Isso é ilustrado na Figura 4. Note que se o contradomínio fosse $\{1, 2, 3, 4\}$, então f não seria sobrejetiva. ◀

EXEMPLO 12 A função $f(x) = x^2$ a partir do conjunto de números inteiros para o conjunto dos números inteiros é sobrejetiva?

Solução: A função f não é sobrejetiva porque não há número inteiro x com $x^2 = -1$, por exemplo. ◀

EXEMPLO 13 A função $f(x) = x + 1$ do conjunto de números inteiros para o conjuntos dos números inteiros é sobrejetiva?

Solução: Essa função é sobrejetiva, porque para cada número inteiro y há um número inteiro x, tal que $f(x) = y$. Para ver isso, note que $f(x) = y$ se e somente se $x + 1 = y$, que se mantém se e somente se $x = y - 1$. ◀

DEFINIÇÃO 8 A função f é *bijetora*, ou é uma *correspondência um para um,* se for injetora e sobrejetora.

Os exemplos 14 e 15 ilustram o conceito de uma função bijetora.

EXEMPLO 14 Considere f como a função de $\{a, b, c, d\}$ para $\{1, 2, 3, 4\}$ com $f(a) = 4$, $f(b) = 2$, $f(c) = 1$ e $f(d) = 3$. A função f é bijetora?

Solução: A função f é injetora e sobrejetiva. É injetora porque dois valores do domínio não têm a mesma imagem. É sobrejetiva porque todos os quatro elementos do contradomínio são imagens de elementos no domínio. Assim, f é uma bijetora. ◀

A Figura 5 dispõe quatro funções, em que a primeira é injetora, mas não sobrejetora, a segunda é sobrejetora, mas não injetora, a terceira é injetora e sobrejetora e a quarta, nem injetora nem sobrejetora. A quinta correspondência na Figura 5 não é uma função, porque projeta um elemento em dois diferentes.

Suponha que f seja uma função de um conjunto A para ele mesmo. Se A é finito, então f é injetora se e somente se for apenas sobrejetiva. (Temos isto a partir do resultado do Exercício 68, ao final desta seção.) Isto não é necessariamente o caso de A ser infinito (como será mostrado na Seção 2.4).

EXEMPLO 15 Considere A como um conjunto. A *função identidade* em A é a função $\iota_A : A \to A$, em que

$$\iota_A(x) = x$$

para todo $x \in A$. Em outras palavras, a função identidade ι_A é a função que projeta cada elemento nele mesmo. A função ι_A é injetora e sobrejetiva, então, ela é bijetora. (Note que ι é a letra grega iota.) ◀

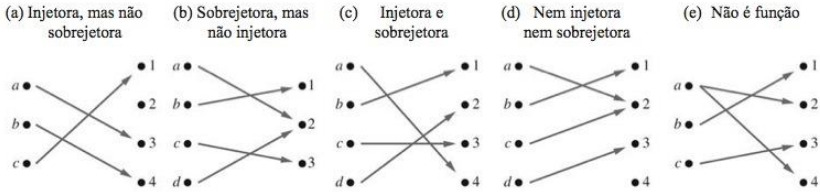


FIGURA 5 **Exemplos de Tipos Diferentes de Correspondências.**

Funções Inversas e Composições de Funções

Considere agora uma função f bijetora do conjunto A para o conjunto B. Como f é uma função sobrejetiva, todo elemento de B é a imagem de algum elemento em A. Além disso, como f é também uma função injetora, todo elemento de B é a imagem de um *único* elemento de A. Conseqüentemente, podemos definir uma nova função de B para A que inverte a correspondência dada por f. Isto nos leva à Definição 9.

DEFINIÇÃO 9 Seja f uma função bijetora do conjunto A para o conjunto B. A *função inversa* de f é a função que leva a um elemento b pertencente a B o único elemento a em A, tal que $f(a) = b$. A função inversa de f é indicada por f^{-1}. Assim, $f^{-1}(b) = a$ quando $f(a) = b$.

Lembre-se: Assegure-se de não confundir a função f^{-1} com a função $1/f$, que é a função que toma para cada x do domínio o valor $1/f(x)$. Note que esta última só faz sentido quando $f(x)$ é um número real diferente de zero. Essas duas funções f^{-1} e $1/f$ não são iguais.

A Figura 6 ilustra o conceito de uma função inversa.

Se uma função f não é uma bijeção, não podemos definir uma função inversa de f. Quando f não é uma bijeção, ela não é injetora ou não é sobrejetora. Se f não é injetora, algum elemento b do contradomínio é a imagem de mais de um elemento no domínio. Se f não é sobrejetiva, para algum elemento b do contradomínio, não existe um elemento a no domínio para que $f(a) = b$. Conseqüentemente, se f não é uma bijeção, não podemos determinar para cada elemento b do contradomínio um único elemento a do domínio, tal que $f(a) = b$ (porque para algum b há mais de um elemento a ou não há projeção para a).

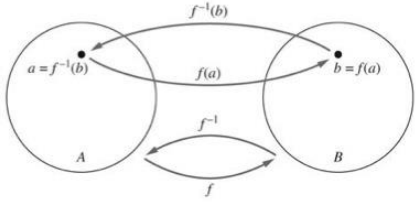


FIGURA 6 **A Função f^{-1} é a Inversa da Função f.**

Uma bijeção é chamada **inversível**, pois podemos definir uma função inversa. Uma função é **não inversível** se não é uma bijeção, porque não existe a função inversa.

EXEMPLO 16 Considere f como a função de $\{a, b, c\}$ para $\{1, 2, 3\}$, tal que $f(a) = 2$, $f(b) = 3$ e $f(c) = 1$. Esta função é inversível? Se for, qual é sua inversa?

Solução: A função f é inversível porque é uma bijeção. A função inversa f^{-1} é dada a partir de f, então $f^{-1}(1) = c$, $f^{-1}(2) = a$ e $f^{-1}(3) = b$. ◀

EXEMPLO 17 Considere $f : \mathbf{Z} \to \mathbf{Z}$ tal que $f(x) = x + 1$. A função f é inversível? Se sim, qual sua função inversa?

Solução: A função f tem uma inversa, porque é uma correspondência um para um, como mostramos antes nesta mesma seção. Para fazer a inversão da correspondência, suponha que y seja a imagem de x, para que $y = x + 1$. Então, $x = y - 1$. Isso significa que $y - 1$ é o único elemento de $\mathbf{Z}$ em que y é projetado por f. Conseqüentemente, $f^{-1}(y) = y - 1$. ◀

EXEMPLO 18 Considere f como a função de $\mathbf{R}$ para $\mathbf{R}$ com $f(x) = x^2$. A função f é inversível?

Solução: Como $f(-2) = f(2) = 4$, f não é injetora. Se uma função inversa fosse definida, teríamos dois elementos como imagem do real 4. Assim, f não é inversível. ◀

Às vezes, podemos restringir o domínio ou o contradomínio de uma função, ou ambos, para obter uma função inversa, como o Exemplo 19 mostra.

EXEMPLO 19 Mostre que se restringirmos a função $f(x) = x^2$ do Exemplo 18 a uma função do conjunto de todos os números reais não negativos para o conjunto de todos os números reais não negativos, então f será inversível.

Solução: A função $f(x) = x^2$ do conjunto dos números reais não negativos para o conjunto dos números reais não negativos é injetora. Para ver isso, note que se $f(x) = f(y)$, então $x^2 = y^2$, assim $x^2 - y^2 = (x + y)(x - y) = 0$. Isto significa que $x + y = 0$ ou $x - y = 0$, assim $x = -y$ ou $x = y$. Como ambos, x e y, são números não negativos, temos que $x = y$. Assim, verifica-se que é uma função injetora. Além disso, $f(x) = x^2$ é sobrejetiva quando o contradomínio é o conjunto de todos os números reais não negativos, pois cada número real não negativo tem uma raiz quadrada. Ou seja, se y for um número real não negativo, existirá um número real não negativo x, tal que $x = \sqrt{y}$, que significa que $x^2 = y$. Como a função $f(x) = x^2$ do conjunto de números reais não negativos para o conjunto dos números reais não negativos é injetora e sobrejetora, então ela é inversível. Sua inversa é dada pela regra $f^{-1}(y) = \sqrt{y}$. ◀

DEFINIÇÃO 10 Considere g como uma função do conjunto A para o conjunto B e considere f como uma função do conjunto B para o conjunto C. A *composição* das funções f e g, indicada por $f \circ g$, é definida por

$$(f \circ g)(a) = f(g(a)).$$

Em outras palavras, $f \circ g$ é a função que determina para o elemento a de A, o elemento determinado por f para $g(a)$. Ou seja, para encontrar $(f \circ g)(a)$, primeiro aplicamos a função g em a para obter $g(a)$ e, então, aplicamos a função f ao resultado $g(a)$ para obter $(f \circ g)(a) = f(g(a))$. Note

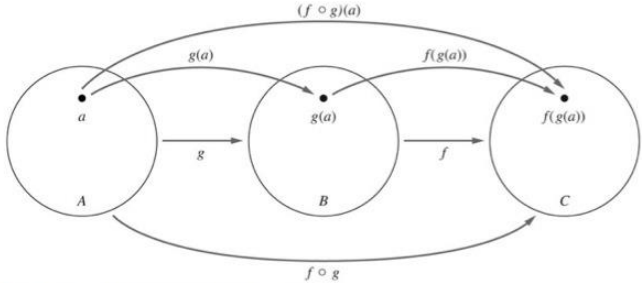


FIGURA 7 **A Composição das Funções *f* e *g*.**

que a composição $f \circ g$ não pode ser definida a menos que o conjunto imagem de g seja um subconjunto do domínio de f. Na Figura 7, é mostrada a composição das funções.

EXEMPLO 20 Considere g como a função do conjunto $\{a, b, c\}$ para ele mesmo, tal que $g(a) = b$, $g(b) = c$ e $g(c) = a$. Considere f como a função do conjunto $\{a, b, c\}$ para o conjunto $\{1, 2, 3\}$, tal que $f(a) = 3, f(b) = 2$ e $f(c) = 1$. Qual é a composição de f e g, e qual a composição de g e f?

Solução: A composição $f \circ g$ é definida por $(f \circ g)(a) = f(g(a)) = f(b) = 2$, $(f \circ g)(b) = f(g(b)) = f(c) = 1$ e $(f \circ g)(c) = f(g(c)) = f(a) = 3$.

Note que $g \circ f$ não é definida, porque o conjunto imagem de f não é um subconjunto do domínio de g. ◀

EXEMPLO 21 Considere f e g como as funções do conjunto dos números inteiros para o conjunto dos números inteiros definidas por $f(x) = 2x + 3$ e $g(x) = 3x + 2$. Qual a composição de f e g? Qual é a composição de g e f?

Solução: Ambas as composições $f \circ g$ e $g \circ f$ são definidas. Além disso,

$$(f \circ g)(x) = f(g(x)) = f(3x + 2) = 2(3x + 2) + 3 = 6x + 7$$

e

$$(g \circ f)(x) = g(f(x)) = g(2x + 3) = 3(2x + 3) + 2 = 6x + 11.$$ ◀

Lembre-se: Note que mesmo $f \circ g$ e $g \circ f$ sendo definidas pelas funções f e g no Exemplo 21, $f \circ g$ e $g \circ f$ não são iguais. Em outras palavras, a propriedade comutativa não se aplica à composição de funções.

Quando fazemos a composição de uma função e sua inversa, em qualquer ordem, a função obtida é a identidade. Para ver isso, suponha que f seja uma bijeção do conjunto A para o conjunto B. Então, a função inversa f^{-1} existe e é uma bijeção de B para A. A função inverte a correspondência da função original, assim $f^{-1}(b) = a$ quando $f(a) = b$, e $f(a) = b$ quando $f^{-1}(b) = a$. Assim,

$$(f^{-1} \circ f)(a) = f^{-1}(f(a)) = f^{-1}(b) = a$$

e

$$(f \circ f^{-1})(b) = f(f^{-1}(b)) = f(a) = b.$$

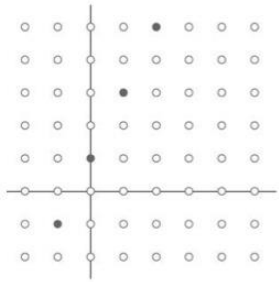


FIGURA 8 O Gráfico de $f(n) = 2n + 1$ de Z para Z.

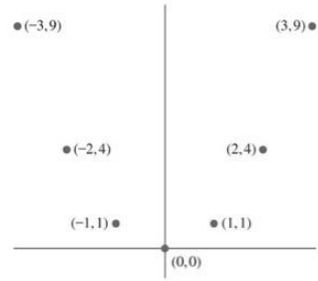


FIGURA 9 O Gráfico de $f(x) = x^2$ de Z para Z.

Conseqüentemente, $f^{-1} \circ f = \iota_A$ de $f \circ f^{-1} = \iota_B$, em que ι_A e ι_B são funções identidade dos conjuntos A e B, respectivamente. Daqui podemos concluir que: $(f^{-1})^{-1} = f$.

Os Gráficos de Funções

Podemos associar um conjunto de pares de $A \times B$ para cada função de A para B. Esse conjunto de pares é chamado de **gráfico** da função e geralmente é disposto pictoricamente como apoio para o entendimento do comportamento da função.

DEFINIÇÃO 11 Considere f como uma função do conjunto A para o conjunto B. O *gráfico* da função f é o conjunto de pares ordenados $\{(a, b) \mid a \in A \text{ e } f(a) = b\}$.

A partir da definição, vemos que o gráfico de uma função f de A para B é o subconjunto $A \times B$ que contém os pares ordenados com a segunda entrada igual ao elemento de B determinado por f para a primeira entrada.

EXEMPLO 22 Esboce o gráfico da função $f(n) = 2n + 1$ do conjunto de números inteiros para o conjunto de números inteiros.

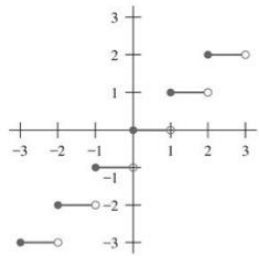
Solução: O gráfico de f é o conjunto de pares ordenados da forma $(n, 2n + 1)$, em que n é um número inteiro. Esse gráfico é mostrado na Figura 8. ◀

EXEMPLO 23 Esboce o gráfico da função $f(x) = x^2$ do conjunto de números inteiros para o conjunto dos números inteiros.

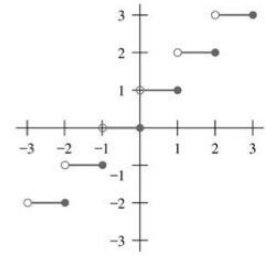
Solução: O gráfico de f é o conjunto de pares ordenados da forma $(x, f(x)) = (x, x^2)$, em que x é um número inteiro. Esse gráfico é mostrado na Figura 9. ◀

Algumas Funções Importantes

Introduziremos agora duas funções importantes em matemática discreta, conhecidas como funções *maior inteiro menor que* e *menor inteiro maior que*. Considere x como um número real. A função maior inteiro menor que leva x para o número inteiro mais próximo menor que ou igual a x, e a função menor inteiro maior que leva x para o número inteiro mais próximo maior que ou igual a x. Essas funções são geralmente usadas quando os objetos são contáveis. Elas desempe-



(a) $y = \lfloor x \rfloor$



(b) $y = \lceil x \rceil$

FIGURA 10 **Gráficos das Funções (a) Maior Inteiro Menor Que e (b) Menor Inteiro Maior Que.**

nham um importante papel na análise de números de passos usados em procedimentos para resolver problemas com tamanhos particulares.

DEFINIÇÃO 12 A *função maior inteiro menor que* (também conhecida como *chão*) determina para o número real x o maior número inteiro menor que ou igual a x. O valor da função maior inteiro menor que em x é indicado por $\lfloor x \rfloor$. A *função menor inteiro maior que* (também conhecida como *teto*) determina para o número real x o menor número inteiro maior que ou igual a x. O valor da função menor inteiro maior que em x é indicado por $\lceil x \rceil$.

Lembre-se: A *função maior inteiro menor que* é também indicada por $[x]$.

EXEMPLO 24 Estes são alguns valores dessas funções:

$$\lfloor \tfrac{1}{2} \rfloor = 0,\ \lceil \tfrac{1}{2} \rceil = 1,\ \lfloor -\tfrac{1}{2} \rfloor = -1,\ \lceil -\tfrac{1}{2} \rceil = 0,\ \lfloor 3{,}1 \rfloor = 3,\ \lceil 3{,}1 \rceil = 4,\ \lfloor 7 \rfloor = 7,\ \lceil 7 \rceil = 7.$$ ◀

Links

Dispusemos os gráficos das funções maior inteiro menor que e menor inteiro maior que na Figura 10. Na Figura 10(a), dispomos o gráfico da função maior inteiro menor que $\lfloor x \rfloor$. Note que essa função tem o mesmo valor entre o intervalo $[n, n + 1)$, isto é, n, e depois vai para $n + 1$ quando $x = n + 1$. Na Figura 10(b), dispomos o gráfico da função menor inteiro maior que $\lceil x \rceil$. Note que essa função tem o mesmo valor no intervalo $(n, n + 1]$, isto é, $n + 1$, e depois vai para $n + 2$ quando x é maior que $n + 1$.

As funções maior inteiro menor que e menor inteiro maior que são úteis em uma grande variedade de aplicações, incluindo aquelas que envolvem armazenamento e transmissão de dados. Considere os exemplos 25 e 26, típicos de cálculos básicos feitos quando os problemas de base de dados e comunicação de dados são estudados.

EXEMPLO 25 O armazenamento de dados em um disco rígido ou a transmissão de dados por meio de uma rede são geralmente representados como uma cadeia de bytes. Cada byte é composto de 8 bits. Quantos bytes são necessários para codificar 100 bits de dados?

Solução: Para determinar o número de bytes necessários, determinamos o menor número inteiro que é maior ou igual ao quociente: 100 por 8, o número de bits em um byte. Conseqüentemente, $\lceil 100/8 \rceil = \lceil 12{,}5 \rceil = 13$ bytes são necessários. ◀

TABELA 1 Propriedades Úteis das Funções Maior Inteiro Menor Que e Menor Inteiro Maior Que. (n é um inteiro)	
(1a)	$\lfloor x \rfloor = n$ se e somente se $n \leq x < n + 1$
(1b)	$\lceil x \rceil = n$ se e somente se $n - 1 < x \leq n$
(1c)	$\lfloor x \rfloor = n$ se e somente se $x - 1 < n \leq x$
(1d)	$\lceil x \rceil = n$ se e somente se $x \leq n < x + 1$
(2)	$x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$
(3a)	$\lfloor -x \rfloor = -\lceil x \rceil$
(3b)	$\lceil -x \rceil = -\lfloor x \rfloor$
(4a)	$\lfloor x + n \rfloor = \lfloor x \rfloor + n$
(4b)	$\lceil x + n \rceil = \lceil x \rceil + n$

EXEMPLO 26 Em um modo de transferência assíncrona (ATM) (um protocolo de comunicação usado em determinadas redes de transmissão), os dados são organizados em células de 53 bytes. Quantas células ATM podem ser transmitidas em 1 minuto em uma conexão que transmite os dados com uma taxa de 500 kilobits por segundo?

Solução: Em 1 minuto, essa conexão pode transmitir $500\,000 \cdot 60 = 30\,000\,000$ bits. Cada célula ATM tem 53 bytes, o que significa que são $53 \cdot 8 = 424$ bits. Para determinar o número de células que podem ser transmitidas em 1 minuto, determinamos o maior inteiro não excedente ao quociente de 30 000 000 por 424. Conseqüentemente, $\lfloor 30\,000\,00/424 \rfloor = 70\,754$ células ATM podem ser transmitidas em 1 minuto com uma conexão de 500 kilobits por segundo. ◀

A Tabela 1, com x indicando um número real, dispõe algumas simples, mas importantes, propriedades dessas duas funções. Como elas aparecem muito freqüentemente em matemática discreta, é útil olhar para essas identidades. Cada propriedade nessa tabela pode ser estabelecida usando as definições das funções. As propriedades (1a), (1b), (1c) e (1d) resultam diretamente das definições. Por exemplo, (1a) estabelece que $\lfloor x \rfloor = n$ se e somente se o número inteiro n for menor que ou igual a x e $n + 1$ for maior que x. Isto é precisamente o que significa para n ser o maior número inteiro não excedente a x, que é a definição de $\lfloor x \rfloor = n$. As propriedades (1b), (1c) e (1d) podem ser estabelecidas de forma semelhante. Demonstraremos a propriedade (4a) usando uma demonstração direta.

Demonstração: Suponha que $\lfloor x \rfloor = m$, em que m é um número inteiro positivo. Pela propriedade (1a), temos que $m \leq x < m + 1$. Adicionando n aos dois lados da inequação, temos que $m + n \leq x + n < m + n + 1$. Usando novamente a propriedade (1a), vemos que $\lfloor x + n \rfloor = m + n = \lfloor x \rfloor + n$. Isto completa a demonstração. As demonstrações das outras propriedades foram deixadas como exercícios. ◁

As funções maior inteiro menor que e menor inteiro maior que são usadas de muitas outras maneiras com propriedades úteis além destas dispostas na Tabela 1. Há também muitas proposições sobre essas funções que podem parecer corretas, mas, na verdade, não são. Consideraremos proposições sobre essas funções nos exemplos 27 e 28.

Uma maneira interessante de considerar as proposições sobre a função maior inteiro menor que é tomar $x = n + \epsilon$, em que $n = \lfloor x \rfloor$ é um número inteiro, e ϵ, a parete de x, que satisfaz a inequação $0 \leq \epsilon < 1$. Da mesma forma, para considerar as proposições sobre a função menor inteiro maior que, é útil escrever $x = n - \epsilon$, em que $n = \lceil x \rceil$ é um número inteiro e $0 \leq \epsilon < 1$.

EXEMPLO 27 Demonstre que se x é um número real, então $\lfloor 2x \rfloor = \lfloor x \rfloor + \lfloor x + \frac{1}{2} \rfloor$.

Exemplos Extras

Solução: Para demonstrar essa proposição, consideremos $x = n + \epsilon$, em que n é um número inteiro positivo e $0 \leq \epsilon < 1$. Há dois casos para considerar, dependendo se ϵ é menor ou maior que ou igual a $\frac{1}{2}$. (A razão de escolhermos esses dois casos ficará clara na demonstração.)

Primeiro consideremos o caso quando $0 \leq \epsilon < \frac{1}{2}$. Nesse caso, $2x = 2n + 2\epsilon$ e $\lfloor 2x \rfloor = 2n$, pois $0 \leq 2\epsilon < 1$. Semelhantemente, $x + \frac{1}{2} = n + (\frac{1}{2} + \epsilon)$, então $\lfloor x + \frac{1}{2} \rfloor = n$, pois $0 < \frac{1}{2} + \epsilon < 1$. Conseqüentemente, $\lfloor 2x \rfloor = 2n$ e $\lfloor x \rfloor + \lfloor x + \frac{1}{2} \rfloor = n + n = 2n$.

Depois, consideremos o caso quando $\frac{1}{2} \leq \epsilon < 1$. Nesse caso, $2x = 2n + 2\epsilon = (2n + 1) + (2\epsilon - 1)$. Como $0 \leq 2\epsilon - 1 < 1$, temos que $\lfloor 2x \rfloor = 2n + 1$. Porque $\lfloor x + \frac{1}{2} \rfloor = \lfloor n + (\frac{1}{2} + \epsilon) \rfloor = \lfloor n + 1 + (\epsilon - \frac{1}{2}) \rfloor$ e $0 \leq \epsilon - \frac{1}{2} < 1$, logo temos que $\lfloor x + \frac{1}{2} \rfloor = n + 1$. Conseqüentemente, $\lfloor 2x \rfloor = 2n + 1$ e $\lfloor x \rfloor + \lfloor x + \frac{1}{2} \rfloor = n + (n + 1) = 2n + 1$. Isto conclui a demonstração. ◀

EXEMPLO 28 Demonstre ou discorde que $\lceil x + y \rceil = \lceil x \rceil + \lceil y \rceil$ para todos os números reais x e y.

Solução: Embora essa proposição possa parecer razoável, ela é falsa. Um exemplo que a contradiz é $x = \frac{1}{2}$ e $y = \frac{1}{2}$. Com esses valores, constatamos que $\lceil x + y \rceil = \lceil \frac{1}{2} + \frac{1}{2} \rceil = \lceil 1 \rceil = 1$, mas $\lceil x \rceil + \lceil y \rceil = \lceil \frac{1}{2} \rceil + \lceil \frac{1}{2} \rceil = 1 + 1 = 2$. ◀

Há certos tipos de funções que serão usadas ao longo deste texto. Estas incluem funções polinomiais, logarítmicas e exponenciais. Uma breve revisão das propriedades dessas funções será necessária neste texto e se encontra no Apêndice 2. Neste livro, a notação $\log x$ será usada para indicar o logaritmo na base 2 de x, pois 2 é a base que usaremos com maior freqüência nos logaritmos. Indicaremos logaritmos na base b, em que b é qualquer número real maior que 1, para $\log_b x$, e o logaritmo natural $\ln x$.

Outra função que será usada ao longo deste texto é a **função fatorial** $f: \mathbf{N} \rightarrow \mathbf{Z}+$, indicada por $f(n) = n!$. O valor de $f(n) = n!$ é o produto dos primeiros números inteiros positivos n, assim $f(n) = 1 \cdot 2 \cdots (n-1) \cdot n$ [e $f(0) = 0! = 1$].

EXEMPLO 29 Temos $f(1) = 1! = 1$, $f(2) = 2! = 1 \cdot 2 = 2$, $f(6) = 6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$ e $f(20) = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10 \cdot 11 \cdot 12 \cdot 13 \cdot 14 \cdot 15 \cdot 16 \cdot 17 \cdot 18 \cdot 19 \cdot 20 = 2\,432\,902\,008\,176\,640\,000$. ◀

Links

JAMES STIRLING (1692–1770) James Stirling nasceu perto da cidade de Stirling, na Escócia. Sua família apoiou fortemente a causa jacobina dos Stuarts como uma alternativa à coroa inglesa. A primeira informação conhecida sobre James é que ele cursou Balliol College, em Oxford, com uma bolsa de estudos em 1711. Entretanto, ele logo perdeu a bolsa de estudos quando se recusou a prometer fidelidade à coroa inglesa. A primeira rebelião jacobina aconteceu em 1715 e Stirling foi acusado de se relacionar com rebeldes. Ele foi acusado de blasfemar contra o Rei George, mas foi absolvido dessa acusação. Embora não pudesse se graduar em Oxford por causa de suas opiniões políticas, ele permaneceu lá durante muitos anos. Stirling publicou seu primeiro trabalho, que era uma extensão do trabalho de Newton sobre curvas planas, em 1717. Ele viajou para Veneza, onde lhe foi prometida uma cadeira em matemática, um compromisso que, infelizmente, não foi cumprido. No entanto, Stirling permaneceu em Veneza, continuando seu trabalho em matemática. Ele freqüentou a Universidade de Pádua em 1721 e, em 1722, retornou a Glasgow. Stirling aparentemente fugiu da Itália depois de aprender os segredos da indústria de vidro italiana, escapando das tentativas, por parte dos donos dessas indústrias, de matá-lo a fim de proteger seus segredos.

Em 1724, Stirling mudou-se para Londres, permanecendo lá por 10 anos, ensinando matemática e ativamente engajado nas pesquisas. Em 1730, ele publicou a obra *Methodus Differentialis*, seu trabalho mais importante, apresentando resultados de séries infinitas, simulações, interpolação e quadrática. Está nessa obra sua fórmula assintótica para $n!$. Stirling trabalhou também com gravitação e a forma da Terra; ele constatou, mas não provou, que a Terra não é uma esfera, e sim que tem a forma elíptica achatada nos pólos. Stirling retornou para a Escócia em 1735, quando foi indicado como gerente de uma companhia de extração escocesa. Ele foi bem-sucedido nesse cargo e até publicou um artigo sobre ventilação nos poços de extração. Ele continuou sua pesquisa matemática, mas de forma reduzida, durante seus anos na indústria de extração. Stirling também é lembrado pela inspeção do Rio Clyde com o objetivo de criar uma série de represas para torná-lo navegável. Em 1752, os cidadãos de Glasgow presentearam-no com uma chaleira de prata como prêmio por seu trabalho.

O Exemplo 29 ilustra que a função fatorial cresce extremamente rápida quando n aumenta. O crescimento rápido da função fatorial é melhor visto com a fórmula de Stirling, um resultado de matemática de altíssimo nível que diz que $n! \sim \sqrt{2\pi n}(n/e)^n$. Aqui, usamos a notação $f(n) \sim g(n)$, que significa que a razão $f(n)/g(n)$ alcança 1 quando n cresce sem limite superior (ou seja, $\lim_{n\to\infty} f(n)/g(n) = 1$). O símbolo $\sim$ é lido como "é assintótico a" ou "aproxima-se de". A fórmula de Stirling foi assim nomeada em homenagem a James Stirling, um matemático escocês do século XVIII.

Exercícios

1. Por que f não é uma função de **R** para **R,** se
- **a)** $f(x) = 1/x$?
- **b)** $f(x) = \sqrt{x}$?
- **c)** $f(x) = \pm\sqrt{(x^2+1)}$?

2. Determine se f é uma função de **Z** para **R,** se
- **a)** $f(n) = \pm n$.
- **b)** $f(n) = \sqrt{n^2+1}$.
- **c)** $f(n) = 1/(n^2-4)$.

3. Determine se f é uma função de um conjunto de todas as cadeias de bits para o conjunto dos números inteiros, se
- **a)** $f(S)$ é a posição de um bit 0 em S.
- **b)** $f(S)$ é o número de bit 1 em S.
- **c)** $f(S)$ é o menor número inteiro i, tal que o i-ésimo bit de S é 1 e $f(S) = 0$ quando S for uma cadeia vazia, uma cadeia sem bits.

4. Encontre o domínio e o conjunto imagem das funções abaixo. Note que em cada caso, para encontrar o domínio, deve-se determinar o conjunto dos valores designados aos elementos pela função.
- **a)** a função que determina, para cada número inteiro não negativo, seu último dígito
- **b)** a função que determina seu próximo maior número inteiro para um número inteiro positivo
- **c)** a função que determina, para uma cadeia de bits, o número de bit 1 na cadeia
- **d)** a função que determina, para uma cadeia de bits, o número de bits na cadeia

5. Encontre o domínio e o conjunto imagem para cada uma das funções abaixo. Note que em cada caso, para encontrar o domínio, deve-se determinar o conjunto dos valores designados aos elementos pela função.
- **a)** a função que determina, para cada cadeia de bits, o número de bit 1 menos o número de bit 0
- **b)** a função que determina, para cada cadeia de bits, duas vezes o número de zeros na cadeia
- **c)** a função que determina o número de bits restantes, quando uma cadeia de bits for organizada em bytes (que são blocos de 8 bits)
- **d)** a função que determina, para cada número inteiro positivo, o maior quadrado perfeito que não excede esse número inteiro

6. Encontre o domínio e o conjunto imagem para cada uma das funções a seguir.
- **a)** a função que determina, para cada par de números inteiros positivos, o primeiro número inteiro do par
- **b)** a função que determina, para cada número inteiro positivo, seu maior dígito decimal
- **c)** a função que determina, para uma cadeia de bits, o número de uns menos o número de zeros na cadeia
- **d)** a função que determina, para cada número inteiro positivo, o maior número inteiro não excedente à raiz quadrada deste inteiro
- **e)** a função que determina, para uma cadeia de bits, a maior cadeia de uns que pode ser encontrada nela

7. Encontre o domínio e o conjunto imagem das funções abaixo.
- **a)** a função que determina, para cada par de números inteiros positivos, o máximo desses dois inteiros
- **b)** a função que determina, para cada número inteiro positivo, quantos dos dígitos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 não aparecem como dígito decimal do número inteiro
- **c)** a função que determina, para uma cadeia de bits, o número de vezes que o bloco 11 aparece
- **d)** a função que determina, para uma cadeia de bits, a posição numérica do primeiro 1 na cadeia e que determina o valor 0 para uma cadeia de bits formada apenas por 0

8. Encontre os valores abaixo.
- **a)** $\lfloor 1{,}1 \rfloor$
- **b)** $\lceil 1{,}1 \rceil$
- **c)** $\lfloor -0{,}1 \rfloor$
- **d)** $\lceil -0{,}1 \rceil$
- **e)** $\lceil 2{,}99 \rceil$
- **f)** $\lceil -2{,}99 \rceil$
- **g)** $\lfloor \frac{1}{2} + \lceil \frac{1}{2} \rceil \rfloor$
- **h)** $\lceil \lfloor \frac{1}{2} \rfloor + \lceil \frac{1}{2} \rceil + \frac{1}{2} \rceil$

9. Encontre os valores abaixo.
- **a)** $\lceil \frac{3}{4} \rceil$
- **b)** $\lfloor \frac{7}{8} \rfloor$
- **c)** $\lceil -\frac{3}{4} \rceil$
- **d)** $\lfloor -\frac{7}{8} \rfloor$
- **e)** $\lceil 3 \rceil$
- **f)** $\lfloor -1 \rfloor$
- **g)** $\lfloor \frac{1}{2} + \lceil \frac{3}{2} \rceil \rfloor$
- **h)** $\lfloor \frac{1}{2} \cdot \lfloor \frac{5}{2} \rfloor \rfloor$

10. Determine se cada uma das funções abaixo, do conjunto $\{a, b, c, d\}$ para ele mesmo, é injetora.
- **a)** $f(a) = b$, $f(b) = a$, $f(c) = c$, $f(d) = d$
- **b)** $f(a) = b$, $f(b) = b$, $f(c) = d$, $f(d) = c$
- **c)** $f(a) = d$, $f(b) = b$, $f(c) = c$, $f(d) = d$

11. Quais funções do Exercício 10 são sobrejetivas?

12. Determine se cada uma das funções abaixo de **Z** para **Z** é injetora.
- **a)** $f(n) = n - 1$
- **b)** $f(n) = n^2 + 1$
- **c)** $f(n) = n^3$
- **d)** $f(n) = \lceil n/2 \rceil$

13. Quais funções do Exercício 12 são sobrejetivas?

14. Determine se $f: \mathbf{Z} \times \mathbf{Z} \to \mathbf{Z}$ é sobrejetiva, se
a) $f(m, n) = 2m - n$.
b) $f(m, n) = m^2 - n^2$.
c) $f(m, n) = m + n + 1$.
d) $f(m, n) = |m| - |n|$.
e) $f(m, n) = m^2 - 4$.

15. Determine se a função $f: \mathbf{Z} \times \mathbf{Z} \to \mathbf{Z}$ é sobrejetiva, se
a) $f(m, n) = m + n$.
b) $f(m, n) = m^2 + n^2$.
c) $f(m, n) = m$.
d) $f(m, n) = |n|$.
e) $f(m, n) = m - n$.

16. Dê um exemplo de uma função de **N** para **N** que é
a) injetora, mas não sobrejetiva.
b) sobrejetiva, mas não injetora.
c) sobrejetiva e injetora (mas diferente da função identidade).
d) nem injetora nem sobrejetiva.

17. Dê uma fórmula explícita para uma função do conjunto de números inteiros para o conjunto de números inteiros positivos, que é
a) injetora, mas não sobrejetora.
b) sobrejetora, mas não injetora.
c) sobrejetora e injetora.
d) nem injetora nem sobrejetora.

18. Determine se cada uma das funções abaixo é uma bijetora de **R** para **R.**
a) $f(x) = -3x + 4$
b) $f(x) = -3x^2 + 7$
c) $f(x) = (x + 1)/(x + 2)$
d) $f(x) = x^5 + 1$

19. Determine se cada uma das funções abaixo é uma bijetora de **R** para **R.**
a) $f(x) = 2x + 1$ **b)** $f(x) = x^2 + 1$
c) $f(x) = x^3$ **d)** $f(x) = (x^2 + 1)/(x^2 + 2)$

20. Considere $f: \mathbf{R} \to \mathbf{R}$ e considere $f(x) > 0$ para todo $x \in \mathbf{R}$. Mostre que $f(x)$ é estritamente crescente se e somente se a função $g(x) = 1/f(x)$ é estritamente decrescente.

21. Considere $f: \mathbf{R} \to \mathbf{R}$ e considere $f(x) > 0$. Mostre que $f(x)$ é estritamente decrescente se e somente se a função $g(x) = 1/f(x)$ é estritamente crescente.

22. Dê um exemplo de uma função crescente com o conjunto de números reais como seu domínio e contradomínio que não seja injetora.

23. Dê um exemplo de uma função decrescente com o conjunto de números reais como seu domínio e contradomínio que não seja injetora.

24. Mostre que a função $f(x) = e^x$ do conjunto dos números reais para o conjunto dos números reais não é inversível, mas se o contradomínio for restrito ao conjunto dos números reais positivos, a função resultante será inversível.

25. Mostre que a função $f(x) = |x|$ do conjunto de números reais para o conjunto dos números reais não negativos não é inversível, mas se o domínio é restrito ao conjunto dos números reais não negativos, então a função resultante é inversível.

26. Considere $S = \{-1, 0, 2, 4, 7\}$. Encontre $f(S)$ se
a) $f(x) = 1$. **b)** $f(x) = 2x + 1$.
c) $f(x) = \lfloor x/5 \rfloor$. **d)** $f(x) = \lfloor (x^2 + 1)/3 \rfloor$.

27. Considere $f(x) = \lfloor x^2/3 \rfloor$. Encontre $f(S)$ se
a) $S = \{-2, -1, 0, 1, 2, 3\}$.
b) $S = \{0, 1, 2, 3, 4, 5\}$.
c) $S = \{1, 5, 7, 11\}$.
d) $S = \{2, 6, 10, 14\}$.

28. Considere $f(x) = 2x$. Qual é
a) $f(\mathbf{Z})$? **b)** $f(\mathbf{N})$? c) $f(\mathbf{R})$?

29. Suponha que g seja uma função de A para B e f seja uma função de B para C.
a) Mostre que se f e g são funções injetoras, então $f \circ g$ é também injetora.
b) Mostre que se f e g são funções sobrejetivas, então $f \circ g$ é também sobrejetiva.

***30.** Se f e $f \circ g$ são injetoras, podemos dizer que g é injetora? Justifique sua resposta.

***31.** Se f e $f \circ g$ são sobrejetivas, podemos dizer que g é sobrejetiva? Justifique sua resposta.

32. Encontre $f \circ g$ e $g \circ f$, em que $f(x) = x^2 + 1$ e $g(x) = x + 2$, são funções de **R** para **R.**

33. Encontre $f + g$ e fg para as funções f e g dadas no Exercício 32.

34. Considere $f(x) = ax + b$ e $g(x) = cx + d$, em que a, b, c e d são constantes. Determine um valor para cada constante a, b, c e d para que $f \circ g = g \circ f$ seja verdadeira.

35. Mostre que a função $f(x) = ax + b$ de **R** para **R** é inversível, em que a e b são constantes, com $a \neq 0$, e encontre a função inversa de f.

36. Considere f como uma função do conjunto A para o conjunto B. Considere S e T como subconjuntos de A. Mostre que
a) $f(S \cup T) = f(S) \cup f(T)$.
b) $f(S \cap T) \subseteq f(S) \cap f(T)$.

37. Dê um exemplo para mostrar que a inclusão do item (b) do Exercício 36 deve ser proposta.

Considere f como uma função do conjunto A para o conjunto B. Considere S como um subconjunto de B. Definimos a **imagem inversa** de S como o subconjunto de A cujos elementos são precisamente todas as pré-imagens de todos os elementos de S. Indicamos a imagem inversa de S por $f^{-1}(S)$, assim $f^{-1}(S) = \{a \in A \mid f(a) \in S\}$. (*Atenção:* A notação f^{-1} é usada de duas maneiras diferentes. Não confunda a notação introduzida aqui com a notação $f^{-1}(y)$ para o valor de y do inverso da função inversa f. Note também que $f^{-1}(S)$, a imagem inversa do conjunto S, faz sentido para todas as funções f, não apenas para funções inversas.)

38. Considere f como a função de **R** para **R** definida por $f(x) = x^2$. Encontre
a) $f^{-1}(\{1\})$. **b)** $f^{-1}(\{x \mid 0 < x < 1\})$.
c) $f^{-1}(\{x \mid x > 4\})$.

39. Considere $g(x) = \lfloor x \rfloor$. Encontre
a) $g^{-1}(\{0\})$. **b)** $g^{-1}(\{-1, 0, 1\})$.
c) $g^{-1}(\{x \mid 0 < x < 1\})$.

40. Considere f como uma função de A para B. Considere S e T como subconjuntos de B. Mostre que
a) $f^{-1}(S \cup T) = f^{-1}(S) \cup f^{-1}(T)$.
b) $f^{-1}(S \cap T) = f^{-1}(S) \cap f^{-1}(T)$.

41. Considere f como uma função de A para B. Considere S como um subconjunto de B. Mostre que $\overline{f^{-1}(S)} = f^{-1}(\overline{S})$.

42. Mostre que $\lfloor x + \frac{1}{2} \rfloor$ é o número inteiro mais próximo de x, exceto quando x está no meio de dois números inteiros, e neste caso é o maior desses dois inteiros.

43. Mostre que $\lceil x - \frac{1}{2} \rceil$ é o número inteiro mais próximo de x, exceto quando x está no meio de dois números inteiros, e neste caso é o menor desses dois inteiros.

44. Mostre que se x é um número real, então $\lceil x \rceil - \lfloor x \rfloor = 1$ se x não é um número inteiro e $\lceil x \rceil - \lfloor x \rfloor = 0$ se x é um número inteiro.

45. Mostre que se x é um número real, então $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$.

46. Mostre que se x é um número real e m é um número inteiro, então $\lfloor x + m \rfloor = \lfloor x \rfloor + m$.

47. Mostre que se x é um número real e n é um número inteiro, então
a) $x < n$ se e somente se $\lfloor x \rfloor < n$.
b) $n < x$ se e somente se $n < \lceil x \rceil$.

48. Mostre que se x é um número real e n é um número inteiro, então
a) $x \leq n$ se e somente se $\lceil x \rceil \leq n$.
b) $n \leq x$ se e somente se $n \leq \lfloor x \rfloor$.

49. Demonstre que se n é um número inteiro, então $\lfloor n/2 \rfloor = n/2$ se n é par e $(n-1)/2$ se n é ímpar.

50. Demonstre que se x é um número real, então $\lfloor -x \rfloor = -\lceil x \rceil$ e $\lceil -x \rceil = -\lfloor x \rfloor$.

51. A função INT é encontrada em algumas calculadoras, em que $\text{INT}(x) = \lfloor x \rfloor$ quando x é um número real não negativo e $\text{INT}(x) = \lceil x \rceil$ quando x é um número real negativo. Mostre que esta função INT satisfaz a identidade $\text{INT}(-x) = -\text{INT}(x)$.

52. Considere a e b como números reais com $a < b$. Use a função chão e/ou teto para expressar o número de inteiros n que satisfazem a inequação $a \leq n \leq b$.

53. Considere a e b como números reais com $a < b$. Use a função chão e/ou teto para expressar o número de inteiros n que satisfazem a inequação $a < n < b$.

54. Quantos bytes são necessários para codificar n bits de dados, em que n é igual a
a) 4? **b)** 10? **c)** 500? **d)** 3000?

55. Quantos bytes são necessários para codificar n bits de dados, em que n é igual a
a) 7? **b)** 17? **c)** 1001? **d)** 28 800?

56. Quantas células ATM (descrito no Exemplo 26) podem ser transmitidas em 10 segundos com um link operando sob as seguintes taxas?
a) 128 kilobits por segundo (1 kilobit = 1000 bits)
b) 300 kilobits por segundo
c) 1 megabit por segundo (1 megabit = 1 000 000 bits)

57. Dados são transmitidos através de uma determinada rede Ethernet em blocos de 1500 octetos (blocos de 8 bits). Quantos blocos são necessários para transmitir as quantidades de dados abaixo através dessa rede de transmissão? (Note que um byte é um sinônimo para octeto, um kilobyte são 1000 bytes e um megabyte são 1 000 000 bytes.)
a) 150 kilobytes de dados
b) 384 kilobytes de dados
c) 1,544 megabyte de dados
d) 45,3 megabytes de dados

58. Construa o gráfico da função $f(n) = 1 - n^2$, de $\mathbf{Z}$ para $\mathbf{Z}$.

59. Construa o gráfico da função $f(x) = \lfloor 2x \rfloor$, de $\mathbf{R}$ para $\mathbf{R}$.

60. Construa o gráfico da função $f(x) = \lfloor x/2 \rfloor$, de $\mathbf{R}$ para $\mathbf{R}$.

61. Construa o gráfico da função $f(x) = \lfloor x \rfloor + \lfloor x/2 \rfloor$, de $\mathbf{R}$ para $\mathbf{R}$.

62. Construa o gráfico da função $f(x) = \lceil x \rceil + \lfloor x/2 \rfloor$, de $\mathbf{R}$ para $\mathbf{R}$.

63. Construa o gráfico das funções abaixo.
a) $f(x) = \lfloor x + \frac{1}{2} \rfloor$
b) $f(x) = \lfloor 2x + 1 \rfloor$
c) $f(x) = \lceil x/3 \rceil$
d) $f(x) = \lceil 1/x \rceil$
e) $f(x) = \lceil x - 2 \rceil + \lfloor x + 2 \rfloor$
f) $f(x) = \lfloor 2x \rfloor \lceil x/2 \rceil$
g) $f(x) = \lceil \lfloor x - \frac{1}{2} \rfloor + \frac{1}{2} \rceil$

64. Construa o gráfico das funções abaixo.
a) $f(x) = \lceil 3x - 2 \rceil$
b) $f(x) = \lceil 0{,}2x \rceil$
c) $f(x) = \lfloor -1/x \rfloor$
d) $f(x) = \lfloor x^2 \rfloor$
e) $f(x) = \lceil x/2 \rceil \lfloor x/2 \rfloor$
f) $f(x) = \lfloor x/2 \rfloor + \lceil x/2 \rceil$
g) $f(x) = \lfloor 2 \lceil x/2 \rceil + \frac{1}{2} \rfloor$

65. Encontre a função inversa de $f(x) = x^3 + 1$.

66. Suponha que f seja uma função inversível de Y para Z e g seja uma função inversa de X para Y. Mostre que o inverso da composição $f \circ g$ é dado por $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.

67. Considere S como um subconjunto de um conjunto universo U. A **função característica** f_S de S é a função de U para o conjunto $\{0, 1\}$, tal que $f_S(x) = 1$, se x pertencer a S e $f_S(x) = 0$, se x não pertencer a S. Considere A e B como conjuntos. Mostre que para todo x,
a) $f_{A \cap B}(x) = f_A(x) \cdot f_B(x)$
b) $f_{A \cup B}(x) = f_A(x) + f_B(x) - f_A(x) \cdot f_B(x)$
c) $f_{\bar{A}}(x) = 1 - f_A(x)$
d) $f_{A \oplus B}(x) = f_A(x) + f_B(x) - 2 f_A(x) f_B(x)$

68. Suponha que f seja uma função de A para B, em que A e B são conjuntos finitos com $|A| = |B|$. Mostre que f é injetora se e somente se for sobrejetiva.

69. Demonstre ou negue cada uma das proposições abaixo sobre funções chão e teto.
a) $\lceil \lfloor x \rfloor \rceil = \lfloor x \rfloor$ para todo número real x.
b) $\lfloor 2x \rfloor = 2 \lfloor x \rfloor$ sempre que x for um número real.
c) $\lfloor x \rfloor + \lceil y \rceil - \lfloor x + y \rfloor = 0$ ou 1 sempre que x e y forem números reais.
d) $\lfloor xy \rfloor = \lfloor x \rfloor \lfloor y \rfloor$ para todo número real x e y.
e) $\left\lceil \frac{x}{2} \right\rceil = \left\lfloor \frac{x+1}{2} \right\rfloor$ para todo número real x.

70. Demonstre ou negue cada uma das proposições abaixo sobre funções chão e teto.
- **a)** $\lfloor \lfloor x \rfloor \rfloor = \lfloor x \rfloor$ para todo número real x.
- **b)** $\lfloor x + y \rfloor = \lfloor x \rfloor + \lfloor y \rfloor$ para todo número real x e y.
- **c)** $\lceil \lceil x/2 \rceil /2 \rceil = \lceil x/4 \rceil$ para todo número real x.
- **d)** $\lfloor \sqrt{\lfloor x \rfloor} \rfloor = \lfloor \sqrt{x} \rfloor$ para todo número real positivo x.
- **e)** $\lfloor x \rfloor + \lfloor y \rfloor + \lfloor x + y \rfloor \leq \lfloor 2x \rfloor + \lfloor 2y \rfloor$ para todo número real x e y.

71. Demonstre que se x é um número real positivo, então
- **a)** $\lfloor \sqrt{\lfloor x \rfloor} \rfloor = \lfloor \sqrt{x} \rfloor$.
- **b)** $\lceil \sqrt{\lfloor x \rfloor} \rceil = \lceil \sqrt{x} \rceil$.

72. Considere x como um número real. Mostre que $\lfloor 3x \rfloor = \lfloor x \rfloor + \lfloor x + \frac{1}{3} \rfloor + \lfloor x + \frac{2}{3} \rfloor$.

Um programa escrito para avaliar uma função pode não produzir o valor correto da função para todos os elementos do domínio dessa função. Por exemplo, um programa pode não produzir um valor correto, porque o cálculo desse valor pode levar a um laço infinito ou a um overflow. Da mesma forma, em matemática abstrata, queremos freqüentemente discutir funções que estão definidas apenas para um subconjunto de números reais, como $1/x$, $\sqrt{x}$, e arcsen(x). Podemos também querer usar tais noções como na função "da criança mais nova", que é indefinida para um casal que não tem filhos, ou o "instante do nascer do sol", que é indefinido para alguns dias no Círculo Ártico.

Para estudar tais situações, usamos o conceito de uma função parcial. Uma **função parcial** f de um conjunto A para um conjunto B é uma projeção de cada elemento a no subconjunto de A, chamado de **domínio de definição** de f, de um único elemento b em B. Os conjuntos A e B são chamados de **domínio** e **contradomínio** de f, respectivamente. Dizemos que f é **indefinida** para os elementos de A que não estão no domínio de definição de f. Escrevemos $f : A \to B$ para indicar que f é uma função parcial de A para B. (Esta é a mesma notação usada para funções. O contexto no qual a notação é usada determina se f é uma função parcial ou uma função total.) Quando o domínio de definição de f é igual a A, dizemos que f é uma **função completa.**

73. Para cada uma das funções parciais a seguir, determine seu domínio, contradomínio, domínio de definição e o conjunto de valores para os quais é indefinida. Determine também se é uma função completa.
- **a)** $f : \mathbf{Z} \to \mathbf{R}, f(n) = 1/n$
- **b)** $f : \mathbf{Z} \to \mathbf{Z}, f(n) = \lceil n/2 \rceil$
- **c)** $f : \mathbf{Z} \times \mathbf{Z} \to \mathbf{Q}, f(m, n) = m/n$
- **d)** $f : \mathbf{Z} \times \mathbf{Z} \to \mathbf{Z}, f(m, n) = mn$
- **e)** $f : \mathbf{Z} \times \mathbf{Z} \to \mathbf{Z}, f(m, n) = m - n$ se $m > n$

74. **a)** Mostre que uma função parcial de A para B pode ser vista como uma função f^* de A para $B \cup \{u\}$, em que u não é um elemento de B e

$$f^*(a) = \begin{cases} f(a) & \text{se } a \text{ pertence ao domínio} \\ & \text{de definição de } f \\ u & \text{se } f \text{ é indefinida em } a. \end{cases}$$

- **b)** Usando a construção em (a), encontre a função f^* correspondente a cada função parcial no Exercício 73.

☞75. **a)** Mostre que se um conjunto S tem cardinalidade m, em que m é um número inteiro positivo, então há uma bijeção entre S e o conjunto $\{1, 2, \ldots, m\}$.
- **b)** Mostre que se S e T são dois conjuntos, cada um com m elementos, em que m é um número inteiro positivo, então há uma bijeção entre S e T.

***76.** Mostre que um conjunto S é infinito se e somente se houver um subconjunto estrito A de S tal que haja uma bijeção entre A e S.

***77.** Mostre que a função polinomial $f : \mathbf{Z}^+ \times \mathbf{Z}^+ \to \mathbf{Z}^+$ com $f(m, n) = (m + n - 2)(m + n - 1)/2 + m$ é injetora e sobrejetiva.

***78.** Mostre que quando você substitui $(3n + 1)^2$ por ocorrência de n e $(3m + 1)^2$ por ocorrência de m do lado direito da fórmula para a função $f(m, n)$ do Exercício 77, você obtém uma função injetora e polinomial $\mathbf{Z} \times \mathbf{Z} \to \mathbf{Z}$. É uma questão aberta sempre que houver uma função injetora e polinomial $\mathbf{Q} \times \mathbf{Q} \to \mathbf{Q}$.

2.4 Seqüências e Somatórios

Introdução

Seqüências são listas ordenadas de elementos. As seqüências são usadas em matemática discreta de várias maneiras. Elas podem ser usadas para representar soluções de certos problemas de contagem, como veremos no Capítulo 7. Elas são também uma estrutura de dados importante em ciência da computação. Esta seção contém uma revisão da notação usada para representar seqüências e somas de termos das seqüências.

Quando os elementos de um conjunto infinito podem ser listados, o conjunto é chamado de contável. Discutiremos nesta seção sobre os conjuntos contáveis e incontáveis. Demonstraremos que o conjunto dos números racionais é contável, mas o conjunto dos números reais não o é.

Seqüências

Uma seqüência é uma estrutura discreta usada para representar uma lista ordenada. Por exemplo, 1, 2, 3, 5, 8 é uma seqüência com cinco termos e 1, 3, 9, 27, 81 , . . . , 30, . . . é uma seqüência infinita.

DEFINIÇÃO 1 Uma *seqüência* é uma função de um subconjunto do conjunto dos números inteiros (geralmente ou do conjunto $\{0, 1, 2, \ldots\}$ ou do conjunto $\{1, 2, 3, \ldots\}$) para um conjunto S. Usamos a notação a_n para indicar a imagem do número inteiro n. Chamamos a_n de *termo* da seqüência.

Usamos a notação $\{a_n\}$ para descrever a seqüência. (Note que a_n representa um único termo da seqüência $\{a_n\}$. Note também que a notação $\{a_n\}$ para a seqüência é a mesma para a notação de um conjunto. Entretanto, o contexto no qual usamos essa notação sempre deixará claro quando estamos trabalhando com conjuntos ou com seqüências. Note que, apesar de termos usado a letra a na notação para uma seqüência, outras letras ou expressões podem ser usadas, dependendo da seqüência que considerarmos. Ou seja, a escolha da letra a é arbitrária.)

Descrevemos as seqüências pela listagem de termos da seqüência em ordem crescente de subscritos.

EXEMPLO 1 Considere a seqüência $\{a_n\}$, em que

$$a_n = \frac{1}{n}.$$

A lista de termos dessa seqüência, começando por a_1, ou seja,

$$a_1, a_2, a_3, a_4, \ldots,$$

começa com

$$1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$$

DEFINIÇÃO 2 Uma *progressão geométrica* é uma seqüência na forma

$$a, ar, ar^2, \ldots, ar^n, \ldots$$

em que o *termo inicial* a e a *razão* r são números reais.

Lembre-se: Uma progressão geométrica é um análogo discreto de uma função exponencial $f(x) = ar^x$.

EXEMPLO 2 As seqüências $\{b_n\}$, com $b_n = (-1)^n$, $\{c_n\}$, com $c_n = 2 \cdot 5^n$ e $\{d_n\}$, com $d_n = 6 \cdot (1/3)^n$ são progressões geométricas com termo inicial e razão comum igual a 1 e -1; 2 e 5; e 6 e 1/3, respectivamente, se começarmos com $n = 0$. A lista de termos $b_0, b_1, b_2, b_3, b_4, \ldots$ começa com

$$1, -1, 1, -1, 1, \ldots;$$

a lista de termos $c_0, c_1, c_2, c_3, c_4, \ldots$ começa com

$$2, 10, 50, 250, 1250, \ldots;$$

e a lista de termos $d_0, d_1, d_2, d_3, d_4, \ldots$ começa com

$$6, 2, \frac{2}{3}, \frac{2}{9}, \frac{2}{27}, \ldots$$

◀

DEFINIÇÃO 3 Uma *progressão aritmética* é uma seqüência da forma

$$a, a + d, a + 2d, \ldots, a + nd, \ldots$$

em que o *termo inicial* a e a *diferença (ou razão)* d são números reais.

Lembre-se: Uma progressão aritmética é um análogo discreto da função linear $f(x) = dx + a$.

EXEMPLO 3 As seqüências $\{s_n\}$, com $s_n = -1 + 4n$ e $\{t_n\}$, com $t_n = 7 - 3n$ são progressões aritméticas com termos iniciais e diferenças comuns iguais a -1 e 4 e 7 e -3, respectivamente, se começarmos em $n = 0$. A lista de termos $s_0, s_1, s_2, s_3, \ldots$ começa com

$$-1, 3, 7, 11, \ldots,$$

e a lista de termos $t_0, t_1, t_2, t_3, \ldots$ começa com

$$7, 4, 1, -2, \ldots.$$

◀

As seqüências na forma $a_1, a_2, \ldots, a_n$ são geralmente usadas em ciência da computação. Essas seqüências finitas são também chamadas de **cadeias.** Esta cadeia é também indicada por $a_1a_2 \ldots a_n$. (Lembre-se que cadeias de bits, que são seqüências finitas de bits, foram introduzidas na Seção 1.1.) A **extensão** da cadeia S é o número de termos nessa cadeia. A **cadeia vazia,** indicada por λ, é a cadeia que não tem termos. A cadeia vazia tem extensão zero.

EXEMPLO 4 A cadeia *abcd* é uma cadeia de extensão quatro. ◀

Seqüências de Números Inteiros Especiais

Um problema comum em matemática discreta é encontrar uma fórmula ou uma regra para construir termos de uma seqüência. Às vezes, apenas poucos termos de uma seqüência que resolvem um problema são conhecidos; o objetivo é também identificar a seqüência. Mesmo que os termos iniciais de uma seqüência não determinem a seqüência inteira (afinal, há infinitas seqüências que começam com o mesmo conjunto finito de termos iniciais), conhecer os primeiros termos pode ajudar você a montar uma conjectura sobre sua seqüência. Uma vez feita essa conjectura, você pode tentar verificar se montou uma seqüência correta.

Ao tentar deduzir uma fórmula possível ou regra para os termos de uma seqüência a partir dos termos iniciais, tente encontrar um padrão desses termos. Você pode também ver se é possível determinar como um termo pode ser produzido a partir de seu antecedente. Há muitas questões que você poderia fazer, mas algumas das mais úteis são:

- Existem séries com o mesmo valor? Ou seja, o mesmo valor ocorre várias vezes em uma fila?
- Existem termos obtidos a partir de termos antecedentes pela adição do mesmo valor ou de algum valor que depende da posição na seqüência?
- Existem termos obtidos a partir de termos antecedentes pela multiplicação de um determinado valor?

- Existem termos obtidos pela combinação, de certa maneira, de termos antecedentes?
- Existem ciclos entre os termos?

EXEMPLO 5 Encontre as fórmulas para as seqüências com os seguintes termos: (a) 1, 1/2, 1/4, 1/8, 1/16 (b) 1, 3, 5, 7, 9 (c) 1, −1, 1, −1, 1.

Exemplos Extras

Solução: (a) Identificamos os denominadores de potência 2. A seqüência com $a_n = 1/2^n$, $n = 0, 1, 2, \ldots$ é uma possibilidade. Essa seqüência proposta é uma progressão geométrica com $a = 1$ e $r = 1/2$.

(b) Notamos que cada termo é obtido pela adição de 2 ao termo anterior. A seqüência com $a_n = 2n + 1$, $n = 0, 1, 2, \ldots$ é uma possibilidade. Essa seqüência proposta é uma progressão aritmética com $a = 1$ e $d = 2$.

(c) Os termos alternam-se entre 1 e −1. A seqüência com $a_n = (-1)^n$, $n = 0, 1, 2 \ldots$ é uma possibilidade. Essa seqüência proposta é uma progressão geométrica com $a = 1$ e $r = -1$. ◀

Os exemplos 6 e 7 mostram como podemos analisar seqüências para descobrir como os termos são construídos.

EXEMPLO 6 Como podemos construir os termos de uma seqüência se os primeiros 10 termos são 1, 2, 2, 3, 3, 3, 4, 4, 4, 4?

Solução: Note que o número inteiro 1 aparece uma vez, o número inteiro 2 aparece duas vezes, o número inteiro 3 aparece três vezes e o número inteiro 4 aparece quatro vezes. Um regra razoável para a construção dessa seqüência é que o número inteiro n aparece exatamente n vezes, então os próximos cinco termos da seqüência seriam 5, os seguintes seis termos seriam 6, e assim por diante. A seqüência construída dessa maneira é uma possibilidade. ◀

EXEMPLO 7 Como podemos construir os termos de uma seqüência se os primeiros 10 termos são 5, 11, 17, 23, 29, 35, 41, 47, 53, 59?

Solução: Note que cada um dos 10 primeiros termos dessa seqüência a partir do primeiro é obtido pela adição de 6 ao termo subseqüente. (Podemos ver isto conferindo que a diferença entre dois termos consecutivos é 6.) Conseqüentemente, o n-ésimo termo poderia ser construído começando com 5 e adicionando 6 a ele um total de $n - 1$ vezes; ou seja, é possível que o n-ésimo termo seja $5 + 6(n - 1) = 6n - 1$. (Essa é uma progressão aritmética com $a = 5$ e $d = 6$.) ◀

Outra técnica útil para encontrar uma regra para a construção de termos de uma seqüência é comparar os termos de uma seqüência de interesse com os termos de uma seqüência de números inteiros conhecida, ou seja, os termos de uma progressão aritmética, termos de uma progressão geométrica, quadrados perfeitos, cubos perfeitos e assim por diante. Os 10 primeiros termos de algumas seqüências que você deve querer guardar estão dispostos na Tabela 1.

EXEMPLO 8 Conjecture uma fórmula simples para a_n, se os 10 primeiros termos da seqüência $\{a_n\}$ são 1, 7, 25, 79, 241, 727, 2185, 6559, 19681, 59047.

Solução: Para resolver este problema, começamos olhando para a diferença entre dois termos consecutivos, mas não vemos um padrão. Quando formamos uma razão de termos consecutivos para vermos se cada termo é um múltiplo do termo anterior, encontramos que essa razão, embora não uma constante, se aproxima de 3. Então, é razoável suspeitar que os termos dessa seqüência são construídos com uma fórmula que envolve 3^n. Comparando esses termos com os termos correspondentes da seqüência $\{3^n\}$, percebemos que o n-ésimo termo é 2 menos a potência de 3 correspondente. Vemos que $a_n = 3^n - 2$ para $1 \leq n \leq 10$ e conjeturamos que esta fórmula é mantida para todo n. ◀

TABELA 1 Algumas Seqüências Usuais.	
n-ésimo termo	*Primeiros 10 termos*
n^2	1, 4, 9, 16, 25, 36, 49, 64, 81, 100, …
n^3	1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, …
n^4	1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000, …
2^n	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, …
3^n	3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049, …
$n!$	1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, …

Veremos ao longo deste texto que as seqüências de números inteiros aparecem em uma grande variedade de contextos em matemática discreta. As seqüências que temos ou que serão encontradas incluem a seqüência de números primos (Capítulo 3), o número de maneiras de ordenar n objetos discretos (Capítulo 5), o número de movimentos necessários para resolver o famoso quebra-cabeça da Torre de Hanói com n discos (Capítulo 7) e o número de coelhos em uma ilha depois de n meses (Capítulo 7).

As seqüências de números inteiros aparecem em uma grande variedade de áreas além da matemática discreta, incluindo biologia, engenharia, química e física, assim como em quebra-cabeças. Uma grande base de dados com mais de 100 000 seqüências de números inteiros diferentes pode ser encontrada na *On-Line Encyclopedia of Integer Sequences*. Esta base de dados foi criada por Neil Sloane na década de 1960. A última versão impressa dessa base foi publicada em 1995 ([SIPI95]); a enciclopédia atual ocuparia mais de 150 volumes do tamanho da edição de 1995. Novas seqüências foram adicionadas regularmente a essa base de dados. Há também um programa acessível via Internet que você pode usar para encontrar seqüências a partir da enciclopédia que coincidem com os termos iniciais que você possui.

Links

Somatórios

Agora, introduziremos a **notação de somatória.** Começamos por descrever a notação usada para expressar a soma dos termos

$$a_m, a_{m+1}, \ldots, a_n$$

da seqüência $\{a_n\}$. Usamos a notação

$$\sum_{j=m}^{n} a_j, \qquad \sum_{j=m}^{n} a_j, \qquad \text{ou} \qquad \sum_{1 \le j \le n} a_j$$

para representar

$$a_m + a_{m+1} + \cdots + a_n.$$

Aqui, a variável j é chamada de **índice da somatória,** e a escolha da letra j como variável é arbitrária; ou seja, poderíamos usar qualquer outra letra, como i ou k. Ou, na notação,

$$\sum_{j=m}^{n} a_j = \sum_{i=m}^{n} a_i = \sum_{k=m}^{n} a_k.$$

Aqui, o índice da somatória assume todos os números inteiros, começando com seu **menor limite** m e terminando com seu **maior limite** n. A letra maiúscula grega sigma, Σ, é usada para indicar somatória.

As leis usuais da aritmética são aplicadas à somatória. Por exemplo, quando a e b são números reais, temos $\sum_{j=1}^{n}(ax_j + by_j) = a\sum_{y=1}^{n}x_j + b\sum_{j=1}^{n}y_j$, em que $x_1, x_2, \ldots, x_n$ e $y_1, y_2, \ldots, y_n$ são números reais. (Não apresentamos uma demonstração formal desta identidade aqui. Tal demonstração pode ser construída usando a indução matemática, um método de demonstração que introduziremos no Capítulo 4. A demonstração também usa as propriedades comutativas e associativas para adição e a propriedade distributiva de multiplicação sobre adição.)

Daremos alguns exemplos de notação de somatória.

EXEMPLO 9 Expresse a soma dos primeiros 100 termos da seqüência $\{a_n\}$, em que $a_n = 1/n$ para $n = 1, 2, 3, \ldots$.

Exemplos Extras

Solução: O menor limite para o índice da somatória é 1 e o maior limite é 100. Escrevemos a soma como

$$\sum_{j=1}^{100}\frac{1}{j}.$$

◀

EXEMPLO 10 Qual o valor de $\sum_{j=1}^{5}j^2$?

Solução: Temos

$$\begin{aligned}\sum_{j=1}^{5}j^2 &= 1^2 + 2^2 + 3^2 + 4^2 + 5^2\\ &= 1 + 4 + 9 + 16 + 25\\ &= 55.\end{aligned}$$

◀

EXEMPLO 11 Qual o valor de $\sum_{k=4}^{8}(-1)^k$?

Solução: Temos

$$\begin{aligned}\sum_{k=4}^{8}(-1)^k &= (-1)^4 + (-1)^5 + (-1)^6 + (-1)^7 + (-1)^8\\ &= 1 + (-1) + 1 + (-1) + 1\\ &= 1.\end{aligned}$$

◀

Links

NEIL SLOANE (Nascido em 1939) Neil Sloane estudou matemática e engenharia elétrica na Universidade de Melbourne, com uma bolsa de estudos da companhia de telefone do governo australiano. Ele teve muitos trabalhos relacionados com telefonia, como construção de postes telefônicos, em seus trabalhos de férias. Depois de se graduar, ele foi designado para cuidar das redes de transmissão de telefone de custo mínimo na Austrália. Em 1962, foi para os Estados Unidos e estudou engenharia elétrica na Cornell University. Sua tese de Ph.D. foi sobre o que chamamos agora de redes neurais.

Ele conseguiu um emprego nos Laboratórios Bell em 1969, trabalhando em muitas áreas, incluindo criação de rede de transmissão e teoria de codificação. Atualmente trabalha para os Laboratórios AT&T, saindo dos Laboratórios Bell quando a AT&T se separou destes em 1996. Um dos seus problemas favoritos é o **kissing problem,** ou **problema do beijo** (nome que ele cunhou), que pergunta quantas esferas podem ser organizadas em n dimensões para que todas elas toquem uma esfera central de mesmo tamanho. (Em duas dimensões, a resposta é 6, pois 6 moedas podem ser colocadas de modo que elas toquem uma moeda central. Em três dimensões, 12 bolas de bilhar podem ser colocadas para que elas toquem uma bola de bilhar central. Duas bolas de bilhar que se tocam dão um "beijo", o que dá sentido à terminologia "kissing problem" e "kissing number", ou "números de beijos".) Sloane, junto com Andrew Odlyzko, mostrou que em 8 e 24 dimensões os números de beijos otimizados são, respectivamente, 240 e 196.560. Os kissing number são conhecidos nas dimensões 1, 2, 3, 8 e 24, mas não em outras dimensões. Os livros de Sloane incluem *Sphere Packings, Lattices and Groups,* 3. ed., com John Conway; *The Theory of Error-Correcting Codes*, com Jessie MacWilliams; *The Encyclopedia of Integer Sequences*, com Simon Plouffe; e *The Rock-Climbing Guide to New Jersey Crags*, com Paul Nick. Este último livro demonstra seu interesse em escalar montanhas rochosas, incluindo mais de 50 sites sobre escalada em New Jersey.

Às vezes é útil modificar o índice da somatória em uma soma. Isso geralmente é feito quando dois somatórios devem ser adicionados, mas seus índices da somatória não combinam. Ao avaliar um índice de soma, é importante fazer as mudanças apropriadas no somatório correspondente. Isso será ilustrado pelo Exemplo 12.

EXEMPLO 12 Suponha que tenhamos a soma

$$\sum_{j=1}^{5} j^2$$

Exemplos Extras mas queremos o índice da somatória entre 0 e 4 em vez de 1 a 5. Para fazer isso, consideremos $k = j - 1$. Então, o novo índice de soma vai de 0 a 4 e o termo j^2 é dado por $(k + 1)^2$. Assim,

$$\sum_{j=1}^{5} j^2 = \sum_{k=0}^{4} (k+1)^2.$$

É fácil verificar que ambas as somas são $1 + 4 + 9 + 16 + 25 = 55$. ◀

Somas de termos de progressões geométricas são normalmente crescentes (tais somas são chamadas de **séries geométricas**). O Teorema 1 nos dá uma fórmula para a soma dos termos de uma progressão geométrica.

TEOREMA 1 Se a e r são números reais e $r \neq 0$, então

$$\sum_{j=0}^{n} ar^j = \begin{cases} \dfrac{ar^{n+1} - a}{r - 1} & \text{se } r \neq 1 \\ (n+1)a & \text{se } r = 1. \end{cases}$$

Demonstração: Considere

$$S = \sum_{j=0}^{n} ar^j.$$

Para calcular S, primeiro multiplicamos os dois lados da equação por r e então manipulamos a soma resultante da seguinte forma:

$$\begin{aligned}
rS &= r\sum_{j=0}^{n} ar^j && \text{substituindo a fórmula da somatória por } S \\
&= \sum_{j=0}^{n} ar^{j+1} && \text{pela propriedade distributiva} \\
&= \sum_{k=1}^{n+1} ar^k && \text{alterando o índice da somatória, com } k = j + 1 \\
&= \left(\sum_{k=0}^{n} ar^k\right) + (ar^{n+1} - a) && \text{removendo o termo } k = n + 1 \text{ e adicionando o termo } k = 0 \\
&= S + (ar^{n+1} - a) && \text{substituindo } S \text{ pela fórmula da somatória}
\end{aligned}$$

A partir dessas equações, vemos que

$$r\,S = S + (ar^{n+1} - a).$$

Isolando S, mostramos que se $r \neq 1$, então

$$S = \frac{ar^{n+1} - a}{r - 1}.$$

Se $r = 1$, então a soma certamente é igual a $(n + 1)a$. ◁

EXEMPLO 13 Somatórias duplas aparecem em muitos contextos (como na análise de laços agrupados em programas de computador). Um exemplo de somatória dupla é

$$\sum_{i=1}^{4}\sum_{j=1}^{3} ij.$$

Para avaliar a dupla soma, primeiro expanda a somatória interna e então continue computando a somatória externa:

$$\begin{aligned}\sum_{i=1}^{4}\sum_{j=1}^{3} ij &= \sum_{i=1}^{4}(i + 2i + 3i)\\ &= \sum_{i=1}^{4} 6i\\ &= 6 + 12 + 18 + 24 = 60.\end{aligned}$$

◀

Podemos também usar a notação de somatória para adicionar todos os valores de uma função, ou termos de um conjunto indexado, em que o índice da somatória é compatível com todos os valores do conjunto. Ou seja, escrevemos

$$\sum_{s \in S} f(s)$$

para representar a soma dos valores de $f(s)$, para todos os membros s de S.

EXEMPLO 14 Qual o valor de $\sum_{s \in \{0,2,4\}} s$?

Solução: Como $\sum_{s \in \{0,2,4\}} s$ representa a soma de valores de s para todos os membros do conjunto $\{0, 2, 4\}$, temos que

$$\sum_{s \in \{0,2,4\}} s = 0 + 2 + 4 = 6.$$

◀

Certas somas aparecem repetidamente em matemática discreta. Ter um grupo de fórmulas para tais somas pode ser útil; a Tabela 2 fornece uma pequena relação das fórmulas para as somas mais freqüentes.

Demonstramos a primeira fórmula dessa tabela no Teorema 1. As próximas três fórmulas nos dão a soma dos primeiros n números inteiros positivos, a soma de seus quadrados e a soma de seus cubos. Essas três fórmulas podem ser obtidas de muitas maneiras diferentes (por exemplo, veja os exercícios 21 e 22 no final desta seção). Note também que cada uma dessas fórmulas, uma vez conhecidas, pode ser facilmente demonstrada usando a indução matemática, assunto da Seção 4.1. As últimas duas fórmulas da Tabela envolvem séries infinitas e serão discutidas brevemente.

O Exemplo 15 mostra como as fórmulas da Tabela 2 podem ser úteis.

TABELA 2 Algumas Fórmulas Para Somatórios Usuais.

Soma	*Fórmula Fechada*
$\sum_{k=0}^{n} ar^k (r \neq 0)$	$\frac{ar^{n+1} - a}{r-1}, r \neq 1$
$\sum_{k=1}^{n} k$	$\frac{n(n+1)}{2}$
$\sum_{k=1}^{n} k^2$	$\frac{n(n+1)(2n+1)}{6}$
$\sum_{k=1}^{n} k^3$	$\frac{n^2(n+1)^2}{4}$
$\sum_{k=0}^{\infty} x^k, \lvert x \rvert < 1$	$\frac{1}{1-x}$
$\sum_{k=1}^{\infty}, kx^{k-1}, \lvert x \rvert < 1$	$\frac{1}{(1-x)^2}$

EXEMPLO 15 Encontre $\sum_{k=50}^{100} k^2$.

Solução: Primeiro note que como $\sum_{k=1}^{100} k^2 = \sum_{k=1}^{49} k^2 + \sum_{k=50}^{100} k^2$, temos que

$$\sum_{k=50}^{100} k^2 = \sum_{k=1}^{100} k^2 - \sum_{k=1}^{49} k^2.$$

Usando a fórmula $\sum_{k=1}^{n} k^2 = n(n+1)(2n+1)/6$ da Tabela 2, vemos que

$$\sum_{k=50}^{100} k^2 = \frac{100 \cdot 101 \cdot 201}{6} - \frac{49 \cdot 50 \cdot 99}{6} = 338350 - 40425 = 297925. \quad \blacktriangleleft$$

ALGUMAS SÉRIES INFINITAS Embora a maioria dos somatórios neste livro seja finita, as séries infinitas são importantes em algumas partes da matemática discreta. As séries infinitas são geralmente estudadas em um curso de cálculo e mesmo a definição dessas séries requer o uso de cálculo, mas algumas vezes elas aparecem em matemática discreta, pois esta trata de coleções infinitas de elementos discretos. Em particular, em nossos estudos futuros de matemática discreta, encontraremos fórmulas fechadas para as séries infinitas nos exemplos 16 e 17 que serão bastante usuais.

EXEMPLO 16 (*Requer cálculo*) Considere x como um número real com $|x| < 1$. Encontre $\sum_{n=0}^{\infty} x^n$.

Exemplos Extras

Solução: Pelo Teorema 1, com $a = 1$ e $r = x$, vemos que $\sum_{n=0}^{k} x^n = \frac{x^{k+1} - 1}{x - 1}$. Como $|x| < 1$, x^{k+1} tende a 0 quando k tende ao infinito. Temos que

$$\sum_{n=0}^{\infty} x^n = \lim_{k \to \infty} \frac{x^{k+1} - 1}{x - 1} = \frac{-1}{x-1} = \frac{1}{1-x}. \quad \blacktriangleleft$$

Podemos produzir novas fórmulas de somatória a partir de fórmulas conhecidas por diferenciação ou integração.

EXEMPLO 17 (*Requer cálculo*) Diferenciando ambos os lados da equação

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x},$$

pelo Exemplo 16, encontramos que

$$\sum_{k=1}^{\infty} kx^{k-1} = \frac{1}{(1-x)^2}.$$

(Esta diferenciação é válida para $|x| < 1$ por um teorema sobre séries infinitas.) ◀

Cardinalidade

Na Seção 2.1, definimos a cardinalidade de um conjunto finito como o número de elementos no conjunto. Ou seja, a cardinalidade de um conjunto finito nos diz quando dois conjuntos finitos têm o mesmo tamanho, ou quando um é maior que o outro. Podemos estender esta noção aos conjuntos infinitos? Retome o Exercício 75 da Seção 2.3 que afirma que há uma correspondência um para um entre dois conjuntos finitos quaisquer com o mesmo número de elementos. Essa observação permite-nos estender o conceito de cardinalidade a todos os conjuntos, finitos ou infinitos, com a Definição 4.

DEFINIÇÃO 4 Os conjuntos A e B têm a mesma *cardinalidade* se e somente se houver uma bijeção de A para B.

Vamos agora dividir conjuntos infinitos em dois grupos, aqueles com a mesma cardinalidade do conjunto dos números naturais, e aqueles com cardinalidade diferente.

DEFINIÇÃO 5 Um conjunto que é finito ou tem a mesma cardinalidade que o conjunto dos números inteiros positivos é chamado de *contável, ou enumerável.* Um conjunto que não é contável é chamado de *incontável, ou não enumerável.* Quando um conjunto infinito S é contável, indicamos a cardinalidade de S por $\aleph_0$ (em que $\aleph$ é alef, a primeira letra do alfabeto hebraico). Escrevemos $|S| = \aleph_0$ e dizemos que S tem cardinalidade "alef zero".

Daremos agora exemplos de conjuntos enumeráveis e não enumeráveis.

EXEMPLO 18 Mostre que o conjunto dos números inteiros positivos e ímpares é um conjunto enumerável.

Solução: Para mostrar que o conjunto dos números inteiros positivos e ímpares é enumerável, exibiremos uma bijeção entre este conjunto e o conjunto dos números inteiros positivos. Considere a função

$$f(n) = 2n - 1$$

de $\mathbf{Z}^+$ para o conjunto dos números inteiros positivos e ímpares. Mostramos que f é uma correspondência um para um, mostrando que é injetora e sobrejetiva. Para ver que é injetora, suponha que $f(n) = f(m)$. Então, $2n - 1 = 2m - 1$, assim $n = m$. Para ver que é sobrejetiva, suponha que t é um número inteiro positivo e ímpar. Então, t tem 1 a menos que o número inteiro par $2k$, em que k é um número natural. Assim, $t = 2k - 1 = f(k)$. Mostramos esta correspondência um para um na Figura 1. ◀

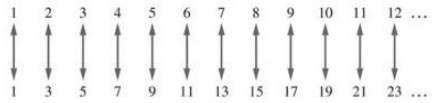


FIGURA 1 Uma Correspondência Um para Um Entre Z^+ e o Conjunto dos Números Inteiros Positivos e Ímpares.

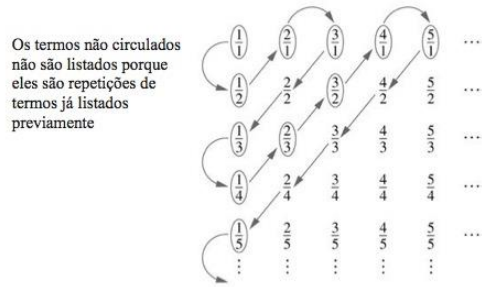


FIGURA 2 Os Números Racionais Positivos são Contáveis.

Um conjunto infinito é contável se e somente se for possível listar os elementos do conjunto em uma seqüência (formada pelos números inteiros positivos). A razão para isto é que a correspondência f um para um do conjunto dos números inteiros positivos para o conjunto S pode ser expressa em termos de uma seqüência $a_1, a_2, \ldots, a_n, \ldots$, em que $a_1 = f(1)$, $a_2 = f(2)$, $\ldots$, $a_n = f(n)$, $\ldots$. Por exemplo, o conjunto dos números inteiros ímpares pode ser listado na seqüência $a_1, a_2, \ldots, a_n, \ldots$, em que $a_n = 2n - 1$.

Podemos mostrar que o conjunto de todos os números inteiros é contável, listando seus membros.

EXEMPLO 19 Mostre que o conjunto de todos os números inteiros é contável.

Solução: Podemos listar todos os números inteiros em uma seqüência, começando com 0 e alternando entre números inteiros positivos e negativos: 0, 1, −1, 2, −2, $\ldots$. Alternadamente, poderíamos encontrar uma correspondência um para um entre o conjunto dos números inteiros positivos e o conjunto de todos os números inteiros. Deixaremos isso para o leitor mostrar que a função $f(n) = n/2$ quando n é par e $f(n) = -(n-1)/2$ quando n é ímpar é tal função. Conseqüentemente, o conjunto de todos os números inteiros é contável. ◀

Não é de surpreender que o conjunto dos números inteiros ímpares e o conjunto de todos os números inteiros são conjuntos contáveis. No entanto, muitas pessoas ficam surpresas ao aprender que o conjunto dos números racionais é contável, como demonstra o Exemplo 20.

EXEMPLO 20 Mostre que o conjunto dos números racionais positivos é contável.

Solução: Pode parecer surpreendente que o conjunto dos números racionais positivos seja contável, mas nós mostraremos como podemos listar os números racionais positivos como uma seqüência $r_1, r_2, \ldots, r_n, \ldots$. Primeiro, note que todo número racional positivo é o quociente p/q de dois números inteiros positivos. Podemos organizar os números racionais positivos listando aqueles com denominador $q = 1$ na primeira linha, aqueles com denominador $q = 2$ na segunda linha e assim por diante, como mostrado na Figura 2.

A maneira de listar os números racionais em uma seqüência é listar primeiro os números racionais positivos p/q com $p + q = 2$, seguido daqueles com $p + q = 3$, seguido daqueles com $p + q = 4$, e assim por diante, seguindo o caminho mostrado na Figura 2. Sempre que passarmos por um número p/q que já foi contado, não o contamos novamente. Por exemplo, quando temos $2/2 = 1$, não o contamos, pois já tivemos $1/1 = 1$. Os termos iniciais na lista de números racionais positivos que construímos são 1, 1/2, 2, 3, 1/3, 1/4, 2/3, 3/2, 4, 5, e assim por diante. Esses números estão circulados, os números não circulados na lista são aqueles que deixamos de lado, pois

já foram listados. Como todos os números racionais positivos são listados uma vez, assim como o leitor pode verificar, mostramos que o conjunto dos números racionais positivos é contável. ◀

Vimos que o conjunto dos números racionais é um conjunto contável. Nós temos um candidato promissor para conjunto incontável? O primeiro lugar que podemos olhar é o conjunto dos números reais. No Exemplo 21, usamos um importante método de demonstração, introduzido em 1879 por Georg Cantor, conhecido como **argumento de diagonalização de Cantor**, para demonstrar que o conjunto dos números reais não é contável.

Links

Este método de demonstração é muito usado em lógica matemática e em teoria da computação.

EXEMPLO 21 Mostre que o conjunto dos números reais é não enumerável.

Exemplos Extras

Solução: Para mostrar que o conjunto dos números reais é não enumerável, suponha que o conjunto dos números reais é enumerável e chegará a uma contradição. Então, supondo que o conjunto dos números reais é enumerável, temos que o subconjunto de todos números reais entre 0 e 1 deve também ser contável (pois qualquer subconjunto de um conjunto contável é também contável; veja o Exercício 36 no final desta seção). Sob esta hipótese, os números reais entre 0 e 1 podem ser listados em alguma ordem, como r_1, r_2, r_3, ... Considere a representação decimal desses números reais como

$$\begin{aligned} r_1 &= 0,d_{11}d_{12}d_{13}d_{14}\ldots \\ r_2 &= 0,d_{21}d_{22}d_{23}d_{24}\ldots \\ r_3 &= 0,d_{31}d_{32}d_{33}d_{34}\ldots \\ r_4 &= 0,d_{41}d_{42}d_{43}d_{44}\ldots \\ &\vdots \end{aligned}$$

em que $d_{ij} \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. (Por exemplo, se $r_1 = 0{,}23794102\ldots$, temos $d_{11} = 2$, $d_{12} = 3$, $d_{13} = 7$, e assim por diante.) Então, forme um novo número real com a expansão decimal $r = 0,d_1d_2d_3d_4\ldots$, em que os dígitos decimais são determinados pela seguinte regra:

$$d_i = \begin{cases} 4 & \text{se } d_{ii} \neq 4 \\ 5 & \text{se } d_{ii} = 4. \end{cases}$$

(Como no exemplo, suponha que $r_1 = 0{,}23794102\ldots$, $r_2 = 0{,}44590138\ldots$, $r_3 = 0{,}09118764\ldots$, $r_4 = 0{,}80553900\ldots$, e assim por diante. Então, temos $r = 0,d_1d_2d_3d_4\ldots = 0{,}4544\ldots$, em que $d_1 = 4$ porque $d_{11} \neq 4$, $d_2 = 5$ porque $d_{22} = 4$, $d_3 = 4$ porque $d_{33} \neq 4$, $d_4 = 4$ porque $d_{44} \neq 4$, e assim por diante.)

Todo número real tem uma única expansão decimal (quando a possibilidade de a expansão ter um final que consiste inteiramente no dígito 9 está excluída). Então, o número real r não é igual a qualquer r_1, r_2, . . . porque a expansão decimal de r difere da expansão decimal de r_i pelo menos no i-ésimo dígito à direita da vírgula na notação decimal, para cada i.

Como há um número real r entre 0 e 1 que não está na lista, a hipótese de que todo número real entre 0 e 1 poderia ser listado é falsa. Assim, todos os números reais entre 0 e 1 não podem ser listados, então o conjunto dos números reais entre 0 e 1 é incontável. Qualquer conjunto com um subconjunto não enumerável é não enumerável (veja o Exercício 37 no final desta seção). Assim, o conjunto dos números reais é não enumerável. ◀

Exercícios

1. Encontre os termos abaixo da seqüência $\{a_n\}$, em que $a_n = 2 \cdot (-3)^n + 5^n$.
a) a_0 **b)** a_1 **c)** a_4 **d)** a_5

2. Qual o termo a_8 da seqüência $\{a_n\}$ se a_n é igual a
a) 2^{n-1}? **b)** 7?
c) $1 + (-1)^n$? **d)** $-(-2)^n$?

3. Quais são os termos a_0, a_1, a_2 e a_3 da seqüência $\{a_n\}$, em que a_n é igual a

a) $2^n + 1$? **b)** $(n+1)^{n+1}$?

c) $\lfloor n/2 \rfloor$? **d)** $\lfloor n/2 \rfloor + \lceil n/2 \rceil$?

4. Quais são os termos a_0, a_1, a_2 e a_3 da seqüência $\{a_n\}$, em que a_n é igual a

a) $(-2)^n$? **b)** 3?

c) $7 + 4^n$? **d)** $2^n + (-2)^n$?

5. Liste os primeiros 10 termos de cada uma das seqüências abaixo.

a) a seqüência que começa com 2 e na qual cada termo sucessivo é o termo anterior acrescido de 3 unidades

b) a seqüência que lista cada número inteiro positivo três vezes, em ordem crescente

c) a seqüência que apresenta os números inteiros positivos e ímpares em ordem crescente, listando cada número inteiro ímpar duas vezes

d) a seqüência cujo n-ésimo termo é $n! - 2^n$

e) a seqüência que começa com 3, na qual cada termo subseqüente é duas vezes o termo anterior

f) a seqüência cujos dois primeiros termos são 1 e cada termo subseqüente é a soma dos dois termos anteriores (Esta é a famosa seqüência de Fibonacci, que estudaremos mais a frente no texto.)

g) a seqüência cujo n-ésimo termo é o número de bits na expansão binária do número n (definida na Seção 3.6)

h) a seqüência em que o n-ésimo termo é o número de letras da palavra em inglês para o índice n

6. Liste os primeiros 10 termos de cada uma das seqüências abaixo.

a) a seqüência obtida começando com 10 e o termo subseqüente obtido pela subtração de 3 do termo anterior

b) a seqüência cujo n-ésimo termo é a soma dos primeiros n números inteiros positivos

c) a seqüência cujo n-ésimo termo é $3^n - 2^n$

d) a seqüência cujo n-ésimo termo é $\lfloor \sqrt{n} \rfloor$

e) a seqüência cujos dois primeiros termos são 1 e 2 e cada termo subseqüente é a soma dos dois termos anteriores

f) a seqüência cujo n-ésimo termo é o maior número inteiro com a expansão binária de n bits (definida na Seção 3.6) (Escreva sua resposta em notação decimal.)

g) a seqüência cujos termos são construídos seqüencialmente por: comece com 1, então adicione 1, depois multiplique por 1, então adicione 2, depois multiplique por 2, e assim por diante

h) a seqüência cujo n-ésimo termo é o maior número inteiro k, tal que $k! \leq n$

7. Encontre pelo menos três seqüências diferentes começando com os termos 1, 2, 4, em que os termos são construídos a partir de um fórmula ou regra simples.

8. Encontre pelo menos três seqüências diferentes começando com os termos 3, 5, 7, em que os termos são construídos a partir de uma fórmula ou regra simples.

9. Para cada uma das listas de números inteiros, forneça uma fórmula ou regra simples para construir os termos de uma seqüência de inteiros que comece com a lista dada. Assumindo que sua fórmula ou regra esteja correta, determine os três próximos termos da seqüência.

a) 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, . . .

b) 1, 2, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, . . .

c) 1, 0, 2, 0, 4, 0, 8, 0, 16, 0, . . .

d) 3, 6, 12, 24, 48, 96, 192, . . .

e) 15, 8, 1, −6, −13, −20, −27, . . .

f) 3, 5, 8, 12, 17, 23, 30, 38, 47, . . .

g) 2, 16, 54, 128, 250, 432, 686, . . .

h) 2, 3, 7, 25, 121, 721, 5041, 40321, . . .

10. Para cada uma das listas de números inteiros, forneça uma fórmula ou regra simples para construir os termos de uma seqüência de inteiros que comece com a lista dada. Assumindo que sua fórmula ou regra esteja correta, determine os três próximos termos da seqüência.

a) 3, 6, 11, 18, 27, 38, 51, 66, 83, 102, . . .

b) 7, 11, 15, 19, 23, 27, 31, 35, 39, 43, . . .

c) 1, 10, 11, 100, 101, 110, 111, 1000, 1001,1010, 1011, . . .

d) 1, 2, 2, 2, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, . . .

e) 0, 2, 8, 26, 80, 242, 728, 2186, 6560, 19682, . . .

f) 1, 3, 15, 105, 945, 10395, 135135, 2027025, 34459425, . . .

g) 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, . . .

h) 2, 4, 16, 256, 65536, 4294967296, . . .

****11.** Mostre que se a_n indica o n-ésimo número inteiro positivo que não é um quadrado perfeito, então $a_n = n + \{\sqrt{n}\}$, em que $\{x\}$ indica o número inteiro mais próximo do número real x.

*** 12.** Considere a_n como o n-ésimo termo da seqüência 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, . . . , construída pela inclusão do número inteiro k exatamente k vezes. Mostre que $a_n = \lfloor \sqrt{2n} + \frac{1}{2} \rfloor$.

13. Quais são os valores das somas abaixo?

a) $\sum_{k=1}^{5}(k+1)$ **b)** $\sum_{j=0}^{4}(-2)^j$

c) $\sum_{i=1}^{10}3$ **d)** $\sum_{j=0}^{8}(2^{j+1} - 2^j)$

14. Quais são os valores das somas abaixo, em que $S = \{1, 3, 5, 7\}$?

a) $\sum_{j \in S} j$ **b)** $\sum_{j \in S} j^2$

c) $\sum_{j \in S}(1/j)$ **d)** $\sum_{j \in S}1$

15. Qual é o valor para cada uma das somas abaixo dos termos de uma progressão geométrica?

a) $\sum_{j=0}^{8}3 \cdot 2^j$ **b)** $\sum_{j=1}^{8}2^j$

c) $\sum_{j=2}^{8}(-3)^j$ **d)** $\sum_{j=0}^{8}2 \cdot (-3)^j$

16. Encontre o valor de cada uma das somas a seguir.

a) $\sum_{j=0}^{8}(1 + (-1)^j)$ **b)** $\sum_{j=0}^{8}(3^j - 2^j)$

c) $\sum_{j=0}^{8}(2 \cdot 3^j + 3 \cdot 2^j)$ **d)** $\sum_{j=0}^{8}(2^{j+1} - 2^j)$

17. Compute cada uma das somas duplas abaixo.

a) $\sum_{i=1}^{2}\sum_{j=1}^{3}(i+j)$ **b)** $\sum_{i=0}^{2}\sum_{j=0}^{3}(2i+3j)$

c) $\sum_{i=1}^{3}\sum_{j=0}^{2} i$ **d)** $\sum_{i=0}^{2}\sum_{j=1}^{3} ij$

18. Compute cada uma das somas duplas abaixo.

a) $\sum_{i=1}^{3}\sum_{j=1}^{2}(i-j)$ **b)** $\sum_{i=0}^{3}\sum_{j=0}^{2}(3i+2j)$

c) $\sum_{i=1}^{3}\sum_{j=0}^{2} j$ **d)** $\sum_{i=0}^{2}\sum_{j=0}^{3} i^2 j^3$

19. Mostre que $\sum_{j=1}^{n}(a_j - a_{j-1}) = a_n - a_0$, em que a_0, a_1, ..., a_n é uma seqüência de números reais. Esse tipo de soma é chamada de **telescópica.**

20. Use a identidade $1/(k(k+1)) = 1/k - 1/(k+1)$ e o Exercício 19 para computar $\sum_{k=1}^{n} 1/(k(k+1))$.

21. Some os dois lados da identidade $k^2 - (k-1)^2 = 2k - 1$ de $k = 1$ a $k = n$ e use o Exercício 19 para encontrar

a) uma fórmula para $\sum_{k=1}^{n}(2k-1)$ (a soma dos primeiros n números naturais ímpares).

b) uma fórmula para $\sum_{k=1}^{n} k$.

*** 22.** Use a técnica dada no Exercício 19, junto com o resultado do Exercício 21b, para derivar a fórmula para $\sum_{k=1}^{n} k^2$ dada na Tabela 2. (*Dica*: Considere $a_k = k^3$ na soma telescópica do Exercício 19.)

23. Encontre $\sum_{k=100}^{200} k$. (Use a Tabela 2.)

24. Encontre $\sum_{k=99}^{200} k^3$. (Use a Tabela 2.)

*** 25.** Encontre uma fórmula para $\sum_{k=0}^{m} \lfloor \sqrt{k} \rfloor$, quando m é um número inteiro positivo.

*** 26.** Encontre uma fórmula para $\sum_{k=0}^{m} \lfloor \sqrt[3]{k} \rfloor$, quando m é um número inteiro positivo.

Há também uma notação especial para produtos. O produto de a_m, a_{m+1}, ..., a_n é representado por

$$\prod_{j=m}^{n} a_j.$$

27. Quais são os valores dos produtos abaixo?

a) $\prod_{i=0}^{10} i$ **b)** $\prod_{i=5}^{8} i$

c) $\prod_{i=1}^{100}(-1)^i$ **d)** $\prod_{i=1}^{10} 2$

Lembre-se que o valor da função fatorial de um número inteiro positivo n, indicado por $n!$, é o produto dos números inteiros positivos de 1 a n. Lembramos também que $0! = 1$.

28. Expresse $n!$ usando a notação de produto.

29. Encontre $\sum_{j=0}^{4} j!$.

30. Encontre $\prod_{j=0}^{4} j!$.

31. Determine se cada um dos conjuntos abaixo é contável ou incontável. Para aqueles que forem contáveis, exiba uma bijeção entre o conjunto dos números naturais e o conjunto

a) dos números inteiros negativos.

b) dos números inteiros pares.

c) dos números reais entre 0 e $\frac{1}{2}$.

d) dos números inteiros que são múltiplos de 7.

32. Determine se cada um dos conjuntos abaixo é contável ou incontável. Para aqueles que forem contáveis, exiba uma bijeção entre o conjunto dos números naturais e o conjunto

a) dos números inteiros maiores que 10.

b) dos números inteiros negativos e ímpares.

c) dos números reais entre 0 e 2.

d) dos números inteiros que são múltiplos de 10.

33. Determine se cada um dos conjuntos abaixo é contável ou incontável. Para aqueles que forem contáveis, exiba uma bijeção entre o conjunto dos números naturais e o conjunto

a) de todas as cadeias de bits que não tenham o bit 0.

b) de todos os números racionais positivos que não podem ser escritos com denominadores menores que 4.

c) dos números reais que não tenha 0 em sua representação decimal.

d) dos números reais que tenha apenas um número finito de 1s em sua representação decimal.

34. Determine se cada um dos conjuntos abaixo é contável ou incontável. Para aqueles que forem contáveis, exiba uma bijeção entre o conjunto dos números naturais e o conjunto

a) dos números inteiros não divisíveis por 3.

b) dos números inteiros divisíveis por 5, mas não por 7.

c) dos números reais com representação decimal consistindo apenas de algarismos 1s

d) dos números reais com representação decimal em que todos os algarismos são 1s ou 9s.

35. Se A é um conjunto incontável e B é um conjunto contável, $A - B$ deverá ser incontável?

36. Mostre que um subconjunto de um conjunto contável é também contável.

37. Mostre que se A e B são conjuntos, A é incontável e $A \subseteq B$, então B é incontável.

38. Mostre que se A e B são conjuntos com a mesma cardinalidade, então o conjunto das partes de A e o conjunto das partes de B têm a mesma cardinalidade.

39. Mostre que se A e B são conjuntos com a mesma cardinalidade e C e D são conjuntos com a mesma cardinalidade, então $A \times C$ e $B \times D$ têm a mesma cardinalidade.

40. Mostre que a união de dois conjuntos contáveis é contável.

**** 41.** Mostre que a união de um número contável de conjuntos contáveis é contável.

42. Mostre que o conjunto $\mathbf{Z}^+ \times \mathbf{Z}^+$ é contável.

***43.** Mostre que o conjunto de todas as cadeias de bits finitas é contável.

***44.** Mostre que o conjunto dos números reais que são soluções de equações quadráticas $ax^2 + bx + c = 0$, em que a, b e c são números inteiros, é contável.

***45.** Mostre que o conjunto de todos os programas computacionais de uma determinada linguagem de programação é contável. [*Dica:* Um programa computacional escrito em uma linguagem de programação pode ser pensado como uma cadeia de símbolos de um alfabeto finito.]

***46.** Mostre que o conjunto de funções do conjunto de números inteiros positivos para o conjunto {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} é incontável. [*Dica:* Primeiro construa uma bijeção entre o conjunto dos números reais entre 0 e 1 e um subconjunto dessas funções. Faça isso pela associação dos números reais $0,d_1d_2 \ldots d_n \ldots$ à função f com $f(n) = d_n$.]

***47.** Dizemos que uma função é **computável** se há um programa de computador que encontra os valores dessa função. Use os exercícios 45 e 46 para mostrar que há funções que não são computáveis.

***48.** Mostre que se S é um conjunto, então não existe uma função sobrejetiva f de S para $P(S)$, o conjunto das partes de S. [*Dica:* Suponha que tal função f exista. Considere $T = \{s \in S \mid s \notin f(s)\}$ e mostre que nenhum elemento s pode existir para $f(s) = T$.]

Termos-chave e Resultados

TERMOS

conjunto: uma coleção de objetos distintos

axioma: uma hipótese básica de uma teoria

paradoxo: uma inconsistência lógica

elemento, membro de um conjunto: um objeto de um conjunto

$\emptyset$ **(conjunto vazio, conjunto nulo):** o conjunto sem elementos

conjunto universo: o conjunto que contém todos os objetos em consideração

diagrama de Venn: uma representação gráfica de um conjunto ou conjuntos

$S = T$ **(igualdade de conjuntos):** S e T têm os mesmos elementos

$S \subseteq T$ **(S é um subconjunto de T):** todo elemento de S também é um elemento de T

$S \subset T$ **(S é um subconjunto estrito de T):** S é um subconjunto de T e $S \neq T$

conjunto finito: um conjunto de n elementos, em que n é um número inteiro não negativo

conjunto infinito: um conjunto que não é finito

$|S|$ **(a cardinalidade de S):** o número de elementos em S

$P(S)$ **(o conjunto de partes de S):** o conjunto de todos os subconjuntos de S

$A \cup B$ **(a união de A e B):** o conjunto que contém aqueles elementos que estão, pelos menos, em um dos conjuntos A e B

$A \cap B$ **(a intersecção de A e B):** o conjunto que contém aqueles elementos que estão em A e em B.

$A - B$ **(a diferença de A e B):** o conjunto que contém aqueles elementos que estão em A, mas não estão em B

$\overline{A}$ **(o complemento de A):** o conjunto de elementos no conjunto universo que não estão em A

$A \oplus B$ **(a diferença simétrica de A e B):** o conjunto que contém aqueles elementos em exatamente um dos conjuntos A e B

tabela de pertinência: uma tabela que mostra a pertinência dos elementos nos conjuntos

função de A para B: uma projeção de um elemento de B para cada elemento de A

domínio de f: o conjunto A, em que f é uma função de A para B

contradomínio de f: o conjunto B, em que f é uma função de A para B

b é a imagem de a em f: $b = f(a)$

a é a pré-imagem de b em f: $f(a) = b$

conjunto imagem de f: o conjunto de imagens de f

função sobrejetiva, sobrejetora: uma função de A para B, tal que todo elemento de B é uma imagem de algum elemento em A

função um para um, injetora: uma função em que todas as imagens dos elementos no domínio são diferentes entre si

correspondência um para um, bijetora: uma função que é injetora e sobrejetiva ao mesmo tempo

inverso de f: a função que inverte a correspondência dada por f (quando f é bijetora)

$f \circ g$ **(composição de f e g):** a função que determina $f(g(x))$ para x

$\lfloor x \rfloor$ **(função maior inteiro menor que):** o maior número inteiro não excedente a x

$\lceil x \rceil$ **(função menor inteiro maior que):** o menor número inteiro maior que ou igual a x

seqüência: uma função com domínio que é um subconjunto do conjunto dos números inteiros

progressão geométrica: uma seqüência na forma $a, ar, ar^2, \ldots,$ em que a e r são números reais

progressão aritmética: uma seqüência na forma $a, a + d, a + 2d, \ldots,$ em que a e d são números reais

cadeia: uma seqüência finita

cadeia vazia: uma cadeia de extensão zero

$\sum_{i=1}^{n} a_i$ **:** a soma de $a_1 + a_2 + \ldots + a_n$

$\prod_{i=1}^{n} a_i$ **:** o produto de $a_1 a_2 \ldots a_n$

conjunto contável, enumerável: um conjunto que é finito ou pode ser colocado em correspondência um para um com o conjunto dos números inteiros positivos

conjunto incontável, não enumerável: um conjunto que não é contável

Argumento de diagonalização de Cantor: uma técnica de demonstração que pode ser usada para mostrar que o conjunto dos números reais é incontável

RESULTADOS

O conjunto identidade dado na Tabela 1 da Seção 2.2

As fórmulas de somatória na Tabela 2 da Seção 2.4

O conjunto dos números racionais é contável.

O conjunto dos números reais é incontável.

Questões de Revisão

1. Explique qual o significado para um conjunto ser um subconjunto de outro conjunto. Como você demonstra que um conjunto é um subconjunto de outro conjunto?
2. O que é conjunto vazio? Mostre que o conjunto vazio é um subconjunto de todo conjunto.
3. **a)** Defina $|S|$, a cardinalidade do conjunto S.
   b) Dê uma fórmula para $|A \cup B|$, em que A e B são conjuntos.
4. **a)** Defina o conjunto de partes de um conjunto S.
   b) Quando o conjunto vazio está no conjunto de partes do conjunto S?
   c) Quantos elementos tem o conjunto de partes de um conjunto S com n elementos?
5. **a)** Defina a união, intersecção, diferença e diferença simétrica de dois conjuntos.
   b) Quais são a união, intersecção, diferença e diferença simétrica do conjunto dos números inteiros positivos e do conjunto dos números inteiros ímpares?
6. **a)** Explique o que significa dois conjuntos serem iguais.
   b) Descreva de quantas maneiras você pode mostrar que dois conjuntos são iguais.
   c) Mostre, de pelo menos duas maneiras diferentes, que os conjuntos $A - (B \cap C)$ e $(A - B) \cup (A - C)$ são iguais.
7. Explique a relação entre equivalências lógicas e identidade de conjuntos.
8. **a)** Defina o domínio, o contradomínio e o conjunto imagem de uma função.
   b) Considere $f(n)$ como a função do conjunto dos números inteiros para o conjunto dos números inteiros, tal que $f(n) = n^2 + 1$. Qual é o domínio, o contradomínio e o conjunto imagem dessa função?
9. **a)** Defina o que significa uma função de um conjunto dos números inteiros positivos para o conjunto dos números inteiros positivos ser injetora.
   b) Defina o que significa uma função do conjunto dos números inteiros positivos para o conjunto dos números inteiros positivos ser sobrejetora.
   c) Dê um exemplo de uma função do conjunto dos números inteiros positivos para o conjunto dos números inteiros positivos que é injetora e sobrejetora.
   d) Dê um exemplo de uma função do conjunto dos números inteiros positivos para o conjunto dos números inteiros positivos que é injetora, mas não é sobrejetora.
   e) Dê um exemplo de uma função do conjunto dos números inteiros positivos para o conjunto dos números inteiros positivos que não é injetora, mas é sobrejetiva.
   f) Dê um exemplo de uma função do conjunto dos números inteiros positivos para o conjunto dos números inteiros positivos que não é injetora nem sobrejetiva.
10. **a)** Defina o inverso de uma função.
    b) Quando é que uma função tem uma inversa?
    c) A função $f(n) = 10 - n$ do conjunto dos números inteiros para o conjunto dos números inteiros tem inversa? Se sim, qual é?
11. **a)** Defina as funções chão e teto do conjunto dos números reais para o conjunto dos números inteiros.
    b) Para quais números reais x é verdade que $\lfloor x \rfloor = \lceil x \rceil$?
12. Faça uma conjectura para criar uma fórmula para os termos da seqüência que começa com 8, 14, 32, 86, 248 e encontre os três próximos termos de sua seqüência.
13. Qual é a soma dos termos da progressão geométrica $a + ar + \ldots + ar^n$, quando $r \neq 1$?
14. Mostre que o conjunto dos números inteiros ímpares é contável.
15. Dê um exemplo de um conjunto incontável.

Exercícios Complementares

1. Considere A como o conjunto das palavras em português que tem a letra x, e B como o conjunto das palavras em português que tem a letra q. Expresse cada um dos conjuntos como uma combinação de A e B.
   a) O conjunto das palavras em português que não tem a letra x.
   b) O conjunto das palavras em português que tem um x e um q.
   c) O conjunto das palavras em português que tem um x, mas não um q.
   d) O conjunto das palavras em português que não tem um x nem um q.
   e) O conjunto das palavras em português que tem um x ou um q, mas não ambos.
2. Mostre que se A é um subconjunto de B, então o conjunto de partes de A é um subconjunto do conjunto de partes de B.
3. Suponha que A e B sejam conjuntos e que o conjunto de partes de A é um subconjunto do conjunto de partes de B. É correto afirmar que A é um subconjunto de B?
4. Considere $\mathbf{E}$ como o conjunto dos números inteiros pares e $\mathbf{O}$ como o conjunto dos números inteiros ímpares. Como padrão, Considere $\mathbf{Z}$ como o conjunto de todos os números inteiros. Determine cada um dos conjuntos abaixo.
   a) $\mathbf{E} \cup \mathbf{O}$ **b)** $\mathbf{E} \cap \mathbf{O}$ **c)** $\mathbf{Z} - \mathbf{E}$ **d)** $\mathbf{Z} - \mathbf{O}$
5. Mostre que se A e B são conjuntos, então $A - (A - B) = A \cap B$.
6. Considere A e B como conjuntos. Mostre que $A \subseteq B$ se e somente se $A \cap B = A$.
7. Considere A, B e C como conjuntos. Mostre que $(A - B) - C$ não é necessariamente igual a $A - (B - C)$.
8. Suponha que A, B e C sejam conjuntos. Demonstre ou discorde que $(A - B) - C = (A - C) - B$.

9. Suponha que A, B, C e D sejam conjuntos. Demonstre ou discorde que $(A - B) - (C - D) = (A - C) - (B - D)$.

10. Mostre que se A e B são conjuntos finitos, então $|A \cap B| \leq |A \cup B|$. Determine quando essa relação é uma igualdade.

11. Considere A e B como conjuntos em um conjunto universo finito U. Liste os conjuntos abaixo em ordem crescente de tamanho.

a) $|A|$, $|A \cup B|$, $|A \cap B|$, $|U|$, $|\emptyset|$

b) $|A - B|$, $|A \oplus B|$, $|A| + |B|$, $|A \cup B|$, $|\emptyset|$

12. Considere A e B como subconjuntos do conjunto universo finito U. Mostre que $|\overline{A} \cap \overline{B}| = |U| - |A| - |B| + |A \cap B|$.

13. Considere f e g como funções de $\{1, 2, 3, 4\}$ para $\{a, b, c, d\}$ e de $\{a, b, c, d\}$ para $\{1, 2, 3, 4\}$, respectivamente, tal que $f(1) = d$, $f(2) = c$, $f(3) = a$ e $f(4) = b$, e $g(a) = 2$, $g(b) = 1$, $g(c) = 3$ e $g(d) = 2$.

a) A função f é injetora? A função g é injetora?

b) A função f é sobrejetiva? A função g é sobrejetiva?

c) As funções f ou g têm uma inversa? Se sim, encontre essa inversa.

14. Considere f como um função injetora do conjunto A para o conjunto B. Considere S e T como subconjuntos de A. Mostre que $f(S \cap T) = f(S) \cap f(T)$.

15. Dê um exemplo para mostrar que a igualdade do Exercício 14 pode não ser verdadeira se f não for injetora.

Suponha que f seja uma função de A para B. Definimos a função S_f de $P(A)$ para $P(B)$ pela regra $S_f(X) = f(X)$ para cada subconjunto X de A. Da mesma forma, definimos a função $S_{f^{-1}}$ de $P(B)$ para $P(A)$ pela regra $S_{f^{-1}}(Y) = f^{-1}(Y)$ para cada subconjunto Y de B. Aqui, estamos usando a Definição 4 e a definição da imagem inversa de um conjunto encontrada no preâmbulo do Exercício 38, ambas da Seção 2.3.

***16.** **a)** Demonstre que se f é uma função injetora de A para B, então S_f é uma função injetora de $P(A)$ para $P(B)$.

b) Demonstre que se f é uma função sobrejetiva de A para B, então S_f é uma função sobrejetiva de $P(A)$ para $P(B)$.

c) Demonstre que se f é uma função sobrejetiva de A para B, então $S_{f^{-1}}$ é uma função injetora de $P(B)$ para $P(A)$.

d) Demonstre que se f é uma função injetora de A para B, então $S_{f^{-1}}$ é uma função sobrejetiva de $P(B)$ para $P(A)$.

e) Use os itens (a) até (d) para concluir que se f é uma bijetora de A para B, então S_f é uma bijetora de $P(A)$ para $P(B)$ e $S_{f^{-1}}$ é bijetora de $P(B)$ para $P(A)$.

17. Demonstre que se f e g são funções de A para B e $S_f = S_g$ (usando a definição do preâmbulo do Exercício 16), então $f(x) = g(x)$ para todo $x \in A$.

18. Mostre que se n é um número inteiro, então $n = \lceil n/2 \rceil + \lfloor n/2 \rfloor$.

19. Para todos os números reais x e y é verdade que $\lfloor x + y \rfloor = \lfloor x \rfloor + \lfloor y \rfloor$?

20. Para todos os números reais x e y é verdade que $\lceil x + y \rceil = \lceil x \rceil + \lceil y \rceil$?

21. Para todos os números reais x e y é verdade que $\lceil x + y \rceil = \lceil x \rceil + \lfloor y \rfloor$?

22. Demonstre que $\lfloor n/2 \rfloor \lceil n/2 \rceil = \lfloor n^2/4 \rfloor$ para todos os números inteiros n.

23. Demonstre que se m é um número inteiro, então $\lfloor x \rfloor + \lfloor m - x \rfloor = m - 1$, a menos que x seja um número inteiro igual a m.

24. Demonstre que se x é um número real, então $\lfloor \lfloor x/2 \rfloor / 2 \rfloor = \lfloor x/4 \rfloor$.

25. Demonstre que se n é um número inteiro ímpar, então $\lceil n^2/4 \rceil = (n^2 + 3)/4$.

26. Demonstre que se m e n são números inteiros positivos e x é um número real, então

$$\left\lfloor \frac{\lfloor x \rfloor + n}{m} \right\rfloor = \left\lfloor \frac{x + n}{m} \right\rfloor.$$

***27.** Demonstre que se m é um número inteiro positivo e x é um número real, então

$$\lfloor mx \rfloor = \lfloor x \rfloor + \left\lfloor x + \frac{1}{m} \right\rfloor + \left\lfloor x + \frac{2}{m} \right\rfloor + \cdots + \left\lfloor x + \frac{m-1}{m} \right\rfloor.$$

***28.** Definimos os **números de Ulam** tomando $u_1 = 1$ e $u_2 = 2$. Além disso, depois de determinar se os números inteiros menores que n são números de Ulam, tomamos n igual ao próximo número de Ulam se este pode ser escrito unicamente como a soma de dois números de Ulam diferentes. Note que $u_3 = 3$, $u_4 = 4$, $u_5 = 6$ e $u_6 = 8$.

a) Encontre os primeiros 20 números de Ulam.

b) Demonstre que existem infinitos números de Ulam.

29. Determine o valor de $\prod_{k=1}^{100} \frac{k+1}{k}$. (A notação usada aqui para produtos é definida no preâmbulo do Exercício 27 da Seção 2.4.)

***30.** Determine uma regra para a construção dos termos da seqüência que começa com 1, 3, 4, 8, 15, 27, 50, 92, . . . , e encontre os próximos quatro termos dessa seqüência.

***31.** Determine uma regra para a construção dos termos da seqüência que começa com 2, 3, 3, 5, 10, 13, 39, 43, 172, 177, 885, 891, . . . , e encontre os próximos quatro termos dessa seqüência.

***32.** Demonstre que se A e B são conjuntos contáveis, então $A \times B$ é também um conjunto contável.

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dados os subconjuntos A e B de um conjunto com n elementos, use cadeias de bits para encontrar $\overline{A}$, $A \cup B$, $A \cap B$, $A - B$ e $A \oplus B$.

2. Dado os multiconjuntos A e B do mesmo conjunto universo, encontre $A \cup B$, $A \cap B$, $A - B$ e $A + B$ (veja o preâmbulo do Exercício 59 da Seção 2.2).

3. Dados os conjuntos fuzzy A e B, encontre $\overline{A}$, $A \cup B$ e $A \cap B$ (veja o preâmbulo do Exercício 61 da Seção 2.2).
4. Dada uma função f de $\{1, 2, \ldots, n\}$ para o conjunto dos números inteiros, determine quando f é injetora.
5. Dada uma função f de $\{1, 2, \ldots, n\}$ para ela mesma, determine quando f é sobrejetiva.
6. Dada uma função bijetora f do conjunto $\{1, 2, \ldots, n\}$ para ele mesmo, encontre f^{-1}.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Dados dois conjuntos finitos, liste todos os elementos do produto cartesiano desses dois conjuntos.
2. Dado um conjunto finito, liste todos os elementos de seu conjunto de partes.
3. Calcule o número de funções injetoras do conjunto S para o conjunto T, em que S e T são conjuntos finitos de vários tamanhos. Você pode determinar uma fórmula para os números de tais funções? (Veremos tal fórmula no Capítulo 5.)
4. Calcule o número de funções sobrejetivas de um conjunto S para um conjunto T, em que S e T são conjuntos finitos de vários tamanhos. Você pode determinar uma fórmula para os números de tais funções? (Veremos tal fórmula no Capítulo 7.)

* 5. Desenvolva uma coleção de regras diferentes para construir termos de uma seqüência e um programa que selecione aleatoriamente uma dessas regras e construa uma determinada seqüência usando tal regra. Faça isso como parte de um programa interativo que leva para o próximo termo da seqüência e determine se a resposta é o próximo termo.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Discuta como um axioma da teoria dos conjuntos pode ser desenvolvido para evitar o paradoxo de Russell. (Veja o Exercício 38 da Seção 2.1.)
2. Pesquise onde o conceito de uma função apareceu pela primeira vez e descreva como esse conceito foi usado primeiramente.
3. Explique as maneiras diferentes de se usar a *Encyclopedia of Integer Sequences*. Descreva também algumas seqüências menos usuais da enciclopédia e como elas aparecem.
4. Defina a seqüência EKG recentemente inventada, descreva algumas de suas propriedades e proponha questões sobre ela.
5. Procure pela definição de um número transcendente. Explique como mostrar que esses números existem e como são construídos. Quais números famosos podem ser mostrados como transcendentes e para quais números famosos ainda é desconhecido se eles são transcendentes?
6. Discuta os números cardinais infinitos e a hipótese do contínuo.

CAPÍTULO

3 Os Fundamentos: Algoritmos, Números Inteiros e Matrizes

3.1 Algoritmos
3.2 O Crescimento de Funções
3.3 Complexidade dos Algoritmos
3.4 Os Números Inteiros e a Divisão
3.5 Os Números Primos e os Máximos Divisores Comuns
3.6 Os Números Inteiros e os Algoritmos
3.7 Aplicações da Teoria dos Números
3.8 Matrizes

Muitos problemas podem ser resolvidos se forem considerados como casos especiais de problemas gerais. Por exemplo, considere o problema de localizar o maior número inteiro na seqüência 101, 12, 144, 212, 98. Este é um caso específico do problema proposto. Para resolver este problema geral, devemos fornecer um algoritmo, que especifica uma seqüência de passos usados para resolvê-lo. Estudaremos algoritmos para resolver diferentes tipos de problemas neste livro. Por exemplo, algoritmos serão desenvolvidos para encontrar o máximo divisor comum de dois números inteiros, para construir todas as ordenações de um conjunto finito, para fazer uma procura em uma lista, para escolher os elementos de uma seqüência e para encontrar o caminho mais curto entre nós em uma rede de transmissão.

Uma consideração importante acerca de um algoritmo é sua complexidade computacional, ou seja, quais são os recursos necessários para usar esse algoritmo na resolução de um problema de determinado tamanho? Para medir a complexidade de algoritmos, usamos as notações O grande e Teta grande, que serão examinadas neste capítulo. Mostraremos também a análise da complexidade de algoritmos. Além disso, discutiremos o que significa a complexidade de um algoritmo na prática e em termos teóricos.

O conjunto de números inteiros tem um papel fundamental em matemática discreta. Em particular, o conceito de divisão de números inteiros é fundamental para a aritmética da computação. Retomaremos brevemente alguns conceitos importantes da teoria dos números, o estudo dos números inteiros e suas propriedades. Alguns algoritmos importantes que envolvem inteiros serão estudados para determinar os máximos divisores comuns, incluindo o algoritmo de Euclides, que foi o primeiro algoritmo descrito há centenas de anos. Os números inteiros podem ser representados usando-se qualquer número inteiro positivo maior que 1 como uma base. Expansões binárias, usadas em ciência da computação, são representadas com uma base 2. Neste capítulo, discutiremos as representações da base b de números inteiros e daremos um algoritmo para encontrá-las. Os algoritmos para aritmética de números inteiros, que foram os primeiros procedimentos a ser chamados de algoritmos, também serão discutidos. Este capítulo introduz também algumas aplicações importantes da teoria dos números. Por exemplo, neste capítulo usaremos essa teoria para fazer mensagens secretas, para construir números pseudo-aleatórios e para determinar a localização da memória para arquivos de computador. A teoria dos números, uma vez considerada como o mais puro dos assuntos, tornou-se um instrumento essencial para a segurança de computadores e da Internet.

As matrizes são usadas em matemática discreta para representar várias estruturas discretas. Retomaremos o material básico sobre matrizes, bem como aritmética matricial, necessário para representar relações e gráficos. A aritmética matricial será usada em vários algoritmos que envolvem essas estruturas.

3.1 Algoritmos

Introdução

Há muitas classes gerais de problemas que aparecem em matemática discreta. Por exemplo: dada uma seqüência de números inteiros, encontre o maior; dado um conjunto, liste todos os seus elementos; dado um conjunto de números inteiros, coloque-os em ordem crescente; dada uma rede de transmissão, encontre o caminho mais curto entre dois vértices. Quando apresentamos problemas como esses, a primeira coisa a fazer é construir um modelo que traduza o problema em um contexto matemático. As estruturas discretas usadas em tais modelos incluem conjuntos, seqüências e funções – estruturas discutidas no Capítulo 2 – assim como outras estruturas como permutações, relações, grafos, árvores, redes e máquinas de estado finito – conceitos que serão discutidos nos próximos capítulos.

Construir o modelo matemático apropriado é apenas parte da solução. Para completá-la, é necessário um método que resolva o problema geral usando esse modelo. Em condições ideais, o que é preciso é um procedimento que siga uma seqüência de passos para chegar à resposta desejada. Tal seqüência de passos é chamada de **algoritmo.**

DEFINIÇÃO 1 Um *algoritmo* é um conjunto finito de instruções precisas para realizar uma operação computacional ou para resolver um problema.

O termo *algoritmo* é a simplificação do nome *al-Khowarizmi*, um matemático do século IX, cujo livro sobre numerais hindus é a base da notação decimal moderna. Originariamente, a palavra *algorismo* era usada nas regras para realizar aritmética usando a notação decimal. *Algorismo* passou a *algoritmo* no século XVIII. Com o crescimento do interesse em máquinas computacionais, o conceito de algoritmo teve seu significado expandido para incluir todos os procedimentos definidos para resolver problemas, não apenas os procedimentos de aritmética. (Discutiremos os algoritmos para realizar a aritmética dos números inteiros na Seção 3.6.)

Neste livro, examinaremos algoritmos que resolvem uma grande variedade de problemas. Nesta seção, usaremos o problema para encontrar o maior número inteiro em uma seqüência de inteiros para mostrar o conceito de algoritmo e as propriedades que ele possui. Descreveremos também algoritmos para localizar um determinado elemento em um conjunto finito. Nas seções subseqüentes, estudaremos procedimentos para encontrar o máximo divisor comum de dois números inteiros, para encontrar o caminho mais curto entre dois pontos em uma rede, para multiplicar matrizes e assim por diante.

EXEMPLO 1 Descreva um algoritmo que possa encontrar o valor máximo (maior) em uma seqüência finita de números inteiros.

Exemplos Extras

Mesmo que o problema de encontrar o maior elemento de uma seqüência seja trivial, ele fornece uma boa ilustração do conceito de um algoritmo. Há também muitos casos em que o maior número inteiro em uma seqüência finita de inteiros é solicitado. Por exemplo, uma universidade pode precisar descobrir a maior nota em uma prova competitiva aplicada a milhares de estudantes. Ou uma organização esportiva pode querer identificar o membro com maior pontuação em um determinado mês. Queremos desenvolver um algoritmo que possa ser usado sempre que aparecer o problema de encontrar o maior elemento em uma seqüência finita de números inteiros.

Podemos especificar um procedimento para resolver esse problema de várias maneiras. Um método simples é usar a língua portuguesa para descrever a seqüência de passos usados. Forneceremos agora essa solução.

Solução do Exemplo 1: Realizamos os seguintes passos.

1. Coloque como máximo temporário o primeiro número inteiro da seqüência. (O máximo temporário é o maior número inteiro examinado em qualquer etapa do processo.)
2. Compare o próximo número inteiro da seqüência com o máximo temporário e, se aquele for maior que este, coloque-o como máximo temporário.

Links

ABU JA'FAR MOHAMMED IBN MUSA AL-KHOWARIZMI (780–850) al-Khowarizmi, astrônomo e matemático, era um membro da Casa da Sabedoria, uma academia de cientistas em Bagdá. O nome al-Khowarizmi significa "da cidade de Kowazizm", que era então parte da Pérsia, mas agora é chamada de *Khiva* e faz parte do território do Uzbequistão. al-Khowarizmi escreveu livros sobre matemática, astronomia e geografia. Os europeus aprenderam álgebra pela primeira vez com seus trabalhos. A palavra *álgebra* vem de al-jabr, parte do título de seu livro *Kitab al-jabr w'al muquabala*. Este livro foi traduzido para o latim e foi amplamente usado. Seu livro sobre o uso dos numerais hindus descreve procedimentos para operações aritméticas que usam esses numerais. Autores europeus usaram uma variante em latim de seu nome, o qual foi mais tarde transformado na palavra *algoritmo,* para descrever os assuntos de aritmética com numerais hindus.

3. Repita o passo anterior se houver mais números inteiros na seqüência.
4. Pare quando não houver mais números inteiros na seqüência. O máximo temporário neste ponto será o maior inteiro da seqüência. ◀

Um algoritmo também pode ser descrito usando-se uma linguagem computacional. Entretanto, quando isso é feito, apenas as instruções permitidas pela linguagem podem ser usadas. Isto geralmente leva a uma descrição de algoritmo que é complexa e difícil de entender. Além disso, como muitas linguagens de programação são de uso comum, seria indesejável escolher uma determinada linguagem. Assim, em vez de usar uma determinada linguagem computacional para especificar algoritmos, uma forma de **pseudocódigo** será usada neste livro. (Todos os algoritmos também serão descritos usando-se a língua portuguesa.) O pseudocódigo fornece um passo intermediário entre a descrição de um algoritmo em língua portuguesa e sua implementação em uma linguagem de programação. Os passos do algoritmo são especificados usando-se instruções parecidas com aquelas usadas em linguagens de programação. Entretanto, em pseudocódigo, as instruções usadas incluem operações bem definidas ou proposições. Um programa de computador pode ser produzido em qualquer linguagem computacional usando a descrição do pseudocódigo como ponto de partida.

O pseudocódigo usado neste livro é livremente baseado na linguagem de programação Pascal. Entretanto, a sintaxe do Pascal, ou de outras linguagens, não será seguida. Além disso, qualquer instrução bem definida pode ser usada neste pseudocódigo. Os detalhes do pseudocódigo utilizado no texto encontram-se no Apêndice 3. O leitor deverá recorrer a este apêndice sempre que houver necessidade.

A descrição de um pseudocódigo do algoritmo para encontrar o maior elemento em uma seqüência finita é dada a seguir.

ALGORITMO 1 Encontrar o Maior Elemento em uma Seqüência Finita.

procedure $max(a_1, a_2, \ldots, a_n$: inteiros)
$max := a_1$
for $i := 2$ **to** n
    if $max < a_i$ **then** $max := a_i$
{max é o maior elemento}

Este algoritmo designa primeiro o termo inicial da seqüência, a_1, para a variável max. O laço "for" é usado para examinar sucessivamente os termos da seqüência. Se um termo for maior que o valor atual de max, ele é designado como o novo valor de max.

Há muitas propriedades que os algoritmos normalmente compartilham. É útil lembrar-se delas quando os algoritmos são descritos. Essas propriedades são:

- *Entrada.* Um algoritmo tem um valor de entrada, ou inicial, de um determinado conjunto.
- *Saída.* Para cada conjunto de valores de entrada, um algoritmo produz valores de saída de um determinado conjunto. Os valores de saída, ou finais, são as soluções para o problema.
- *Certeza.* Os passos de um algoritmo devem ser definidos precisamente.
- *Exatidão.* Um algoritmo deverá produzir os valores finais corretos para cada conjunto de valores iniciais.
- *Finitude.* Um algoritmo deverá produzir o valor final desejado depois de um número de passos finito (possivelmente grande) para as entradas em um conjunto.
- *Efetividade.* Deve ser possível realizar cada passo de um algoritmo de modo exato e em uma quantidade finita de tempo.
- *Generalidade.* O procedimento deverá ter aplicabilidade para todos os problemas da forma desejada, não apenas para um determinado conjunto de valores iniciais.

EXEMPLO 2 Mostre que o Algoritmo 1 utilizado para encontrar o maior elemento de uma seqüência finita de números inteiros tem todas as propriedades listadas.

Solução: A entrada para o Algoritmo 1 é uma seqüência de números inteiros. A saída é o maior número inteiro da seqüência. Cada passo do algoritmo é definido com precisão, porque apenas as tarefas, um laço finito e proposições condicionais aparecem. Para mostrar que o algoritmo está correto, devemos mostrar que quando ele termina, o valor da variável *max* é o maior termo da seqüência. Para verificar isso, observe que o valor inicial de *max* é o primeiro termo da seqüência; como todos os termos sucessivos da seqüência são examinados, *max* é atualizado com o valor de um termo se este exceder o máximo dos termos previamente examinados. Este argumento (informal) mostra que quando todos os termos tiverem sido examinados, *max* será igual ao valor do maior termo. (Uma demonstração rigorosa deste procedimento requer técnicas que serão desenvolvidas na Seção 4.1.) O algoritmo usa um número finito de passos porque ele termina depois que todos os números inteiros da seqüência tiverem sido examinados. O algoritmo pode ser desenvolvido em um tempo finito porque cada passo é ou uma comparação ou uma tarefa. Por fim, o Algoritmo 1 é geral porque pode ser usado para encontrar o valor máximo de qualquer seqüência finita de números inteiros. ◄

Algoritmos de Busca

O problema de localizar um elemento em uma lista ordenada aparece em muitos contextos. Por exemplo, um programa que verifica a ortografia de uma palavra procura-a em um dicionário, que é apenas uma lista ordenada de palavras. Problemas desse tipo são chamados de **problemas de busca.** Discutiremos muitos algoritmos de busca nesta seção. Estudaremos o número de passos usados por esses algoritmos na Seção 3.3.

O problema de busca geral pode ser assim descrito: Localize um elemento x em uma lista de elementos distintos $a_1, a_2, \ldots, a_n$, ou determine que ele não está na lista. A solução para este problema de busca é a localização do termo na lista que é igual a x (ou seja, i é a solução se $x = a_i$) e é 0 se x não estiver na lista.

A BUSCA LINEAR O primeiro algoritmo que apresentaremos é chamado de algoritmo de **busca linear,** ou **busca seqüencial**. O algoritmo de busca linear começa com a comparação de x e a_1. Quando $x = a_1$, a solução é a localização de a_1, ou seja, 1. Quando $x \neq a_1$, comparamos x com a_2. Se $x = a_2$, a solução é a localização de a_2, ou seja, 2. Quando $x \neq a_2$, comparamos x com a_3. Continue esse processo, comparando x sucessivamente com cada termo da lista até que encontre o elemento, pois a solução é a localização desse termo, a menos que ele não seja encontrado. Se a lista toda for analisada sem encontrar a localização de x, a solução é 0. O pseudocódigo para o algoritmo de busca linear é apresentado como Algoritmo 2.

Links

ALGORITMO 2 O Algoritmo de Busca Linear.

procedure *busca linear*(x: integer, $a_1, a_2, \ldots, a_n$: inteiros distintos)
$i := 1$
while ($i \leq n$ e $x \neq a_i$)
    $i := i + 1$
if $i \leq n$ **then** *localização* $:= i$
else *localização* $:= 0$
{*localização* é o subscrito igual a x, ou 0 se x não for encontrado}

A BUSCA BINÁRIA Consideremos agora outro algoritmo de busca. Este algoritmo pode ser usado quando a lista tem termos que aparecem em ordem crescente de tamanho (por exemplo: se os termos forem números, eles estão listados do menor para o maior; se forem palavras, eles são listados em ordem alfabética). Este segundo algoritmo de busca é chamado de **algoritmo de busca binária.** Ele funciona a partir da comparação do elemento localizado no meio da lista. A lista é então dividida em duas sublistas menores de mesmo tamanho, ou uma dessas pequenas listas pode ter menos termos que a outra. A busca continua pela restrição da busca na sublista apropriada com base na comparação do elemento que será colocado como termo central. Na Seção 3.3, será mostrado que o algoritmo de busca binária é muito mais eficiente que o algoritmo de busca linear. O Exemplo 3 demonstra como uma busca binária funciona.

Links

EXEMPLO 3 Para procurar 19 na lista

1 2 3 5 6 7 8 10 12 13 15 16 18 19 20 22,

primeiro divida a lista, que tem 16 termos, em duas listas menores de oito termos, ou seja,

1 2 3 5 6 7 8 10 12 13 15 16 18 19 20 22.

Então, compare 19 com o maior termo da primeira lista. Como $10 < 19$, a procura pelo 19 pode ser restrita à lista que contém do 9º ao 16º termo da lista original. Depois, divida esta lista, que contém oito termos, em duas listas menores de quatro termos cada, ou seja,

12 13 15 16 18 19 20 22.

Como $16 < 19$ (comparando 19 com o maior termo da primeira lista), a pesquisa é restrita à segunda dessas listas, que contém do 13º ao 16º termo da lista original. A lista 18 19 20 22 é dividida em duas listas, ou seja,

18 19 20 22.

Como 19 não é maior que o maior termo da primeira dessas duas listas, que também é 19, a busca é restrita à primeira lista: 18 19, que contém o 13º e o 14º termos da lista original. Depois, essa lista de dois termos é divida em duas listas com um termo cada: 18 e 19. Como $18 < 19$, a busca é restrita à segunda lista: a lista que contém o 14º termo da lista, que é 19. Agora que a pesquisa foi restrita a um termo, fazemos uma comparação, para estarmos certos de ter encontrado o número desejado, e 19 é localizado como 14º termo na lista original. ◀

Especificaremos agora os passos do algoritmo de busca binária. Para buscar o número inteiro x na lista $a_1, a_2, \ldots, a_n$, em que $a_1 < a_2 < \ldots < a_n$, comece comparando x com o termo central da seqüência, a_m, em que $m = \lfloor (n + 1)/2 \rfloor$. (Lembre-se de que $\lfloor x \rfloor$ é o maior número inteiro não excedente a x.) Se $x > a_m$, a pesquisa por x pode ser restrita à segunda metade da seqüência, que é $a_{m+1}, a_{m+2}, \ldots, a_n$. Se x não for maior que a_m, a busca por x pode ser restrita à primeira metade da seqüência, que é $a_1, a_2, \ldots, a_m$.

A pesquisa foi agora restrita à lista com não mais de $\lceil n/2 \rceil$ elementos. (Lembre-se de que $\lceil x \rceil$ é o menor número inteiro maior que ou igual a x.) Usando o mesmo procedimento, compare x ao termo central da lista restrita. Então, restrinja a busca à primeira ou segunda metade dessa lista. Repita esse processo até obter uma lista com um termo. Então, determine se este termo é x. O pseudocódigo para o algoritmo de busca binária é apresentado como Algoritmo 3.

ALGORITMO 3 O Algoritmo de Busca Binária.

procedure *busca binária* (x: integer, $a_1, a_2, \ldots, a_n$: inteiros em ordem crescente)
$i := 1$ {i é o ponto final esquerdo do intervalo da busca}
$j := n$ {j é o ponto final direito do intervalo da busca}
while $i < j$
begin
  $m := \lfloor (i + j)/2 \rfloor$
  if $x > a_m$ **then** $i := m + 1$
  else $j := m$
end
if $x = a_i$ **then** *localização* $:= i$
else *localização* $:= 0$
{*localização* é o subscrito igual a x, ou 0 se x não for encontrado}

O Algoritmo 3 funciona com sucessivas divisões da seqüência para recomeçar a busca. Em qualquer etapa, apenas os termos que começam com a_i e que terminam com a_j são considerados. Em outras palavras, i e j são o menor e o maior índices dos termos restantes, respectivamente. O Algoritmo 3 continua restringindo a seqüência para começar a busca até restar apenas um termo da seqüência. Quando isto ocorre, uma comparação é feita para ver se este termo é igual a x.

Ordenação

Demo

Ordenar os elementos de uma lista é um problema que surge em muitos contextos. Por exemplo, para produzir uma lista telefônica, é necessário colocar em ordem alfabética os nomes dos assinantes. Colocar os endereços em ordem em uma caixa de entrada de e-mails pode determinar se há endereços duplicados. Para criar um dicionário útil, as palavras precisam ser colocadas em ordem alfabética. Da mesma forma, para construir uma lista, devemos, muitas vezes, ordenar os dados de acordo com a parte numérica em ordem crescente.

Suponha que temos uma lista de elementos de um conjunto. Além disso, suponha que temos uma maneira de ordenar os elementos do conjunto. (A noção de ordenar elementos de conjuntos será estudada detalhadamente na Seção 8.6.) **Ordenar** é colocar esses elementos em uma lista, na qual eles aparecem em ordem crescente. Por exemplo, ordenar a lista 7, 2, 1, 4, 5, 9 produz a lista 1, 2, 4, 5, 7, 9. Ordenar a lista d, h, c, a, f (usando ordem alfabética) produz a lista a, c, d, f, h.

Uma grande porcentagem de recursos computacionais é direcionada para ordenar alguma coisa. Assim, muitos esforços foram feitos para desenvolver os algoritmos de ordenação. Um número surpreendente desses algoritmos foi criado usando estratégias diferentes. Em seu trabalho fundamental, *The Art of Computer Programming,* Donald Knuth dedicou 400 páginas aos algoritmos de ordenação, aprofundando o estudo de cerca de 15 algoritmos de ordenação diferentes! Há muitas razões pelas quais esse assunto interessa aos cientistas da computação e matemáticos. Entre elas está o fato de alguns algoritmos serem fáceis de implementar, alguns serem mais eficientes (ou, no geral, quando são dadas entradas com certas características, como as listas levemente fora de ordem), alguns levarem vantagem em determinadas arquiteturas de computação e alguns serem particularmente ágeis. Nesta seção, introduziremos dois algoritmos de ordenação, bubble sort (ordenação por flutuação) e insertion sort (ordenação por inserção). Dois outros algoritmos de ordenação, o selection sort (ordenação por seleção) e o insertion sort binário (ordenação por inserção binária), serão introduzidos nos exercícios no final desta seção, e o shaker sort será introduzido nos Exercícios Complementares no final deste capítulo. Na Seção 4.4, discutiremos merge sort (ordenação por fusão) e introduziremos quick sort (ordenação rápida) nos exercícios desta seção; tournament sort (oordem de torneio) será introduzido no conjunto de exercícios da Seção 10.2. Desenvolveremos os algoritmos de ordenação porque a ordenação é um problema importante e porque alguns desses algoritmos podem servir como exemplos de muitos conceitos importantes.

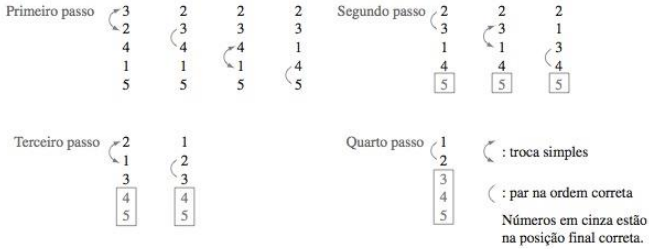


FIGURA 1 Os Passos de um Bubble Sort.

Links

O BUBBLE SORT O **bubble sort** é um dos algoritmos de ordenação mais simples, mas não um dos mais eficientes. Ele coloca uma lista em ordem crescente pela comparação sucessiva de elementos adjacentes, mudando-os se eles estiverem na ordem errada. Para desenvolver o bubble sort, realizamos esta operação básica de trocar um elemento maior por um menor se estiverem em seqüência, começando no início da lista e passando por toda ela. Continuamos com esse procedimento até que a ordenação esteja completa. O pseudocódigo para o bubble sort é dado como Algoritmo 4. Podemos imaginar os elementos da lista colocados em uma coluna. No bubble sort, os elementos menores "flutuam" no topo pela troca com os elementos maiores, que "afundam". Isto é ilustrado no Exemplo 4.

EXEMPLO 4 Use o bubble sort para colocar 3, 2, 4, 1, 5 em ordem crescente.

Solução: Os passos deste algoritmo são ilustrados na Figura 1. Comece comparando os primeiros dois elementos, 3 e 2. Como $3 > 2$, troque o 3 pelo 2, produzindo a lista 2, 3, 4, 1, 5. Como $3 < 4$, continue a comparação dos próximos elementos 4 e 1. Como $4 > 1$, troque o 1 pelo 4, produzindo a lista 2, 3, 1, 4, 5. Como $4 < 5$, o primeiro passo está completo. O primeiro passo garante que o maior elemento, 5, esteja na posição correta.

O segundo passo começa pela comparação entre 2 e 3. Como esses dois estão na ordem correta, 3 e 1 são comparados. Como $3 > 1$, esses números são trocados, produzindo 2, 1, 3, 4, 5. Como $3 < 4$, esses números estão na ordem certa. Não é necessário fazer nenhuma comparação nesta etapa, porque 5 já está na posição certa. O segundo passo garante que os dois maiores elementos, 4 e 5, estejam em suas posições corretas.

O terceiro passo começa pela comparação entre 2 e 1. Eles são trocados, pois $2 > 1$, produzindo 1, 2, 3, 4, 5. Como $2 < 3$, esses dois elementos estão na ordem certa. Não é necessário fazer nenhuma comparação nesta etapa, porque 4 e 5 já estão na posição certa. O terceiro passo garante que os três maiores elementos, 3, 4 e 5, estejam em suas posições corretas.

O quarto passo consiste em uma comparação, ou seja, a comparação entre 1 e 2. Como $1 < 2$, esses elementos estão na ordem correta. Isto completa o bubble sort. ◀

ALGORITMO 4 O Bubble Sort.

procedure *bubblesort*($a_1, \ldots, a_n$: real {números reais com $n \geq 2$})
for $i := 1$ **to** $n - 1$
  for $j := 1$ **to** $n - i$
    if $a_j > a_{j+1}$ **then** troque a_j por a_{j+1}
{$a_1, \ldots, an$ está em ordem crescente}

Links

O INSERTION SORT O **insertion sort** é um algoritmo de ordenação simples, mas geralmente não é o mais eficiente. Para ordenar uma lista com n elementos, o insertion sort começa com o segundo elemento. Ele compara este segundo elemento com o primeiro e coloca-o antes do primeiro elemento se ele não exceder o primeiro elemento, e depois do primeiro elemento se ele o exceder. Neste ponto, os dois primeiros elementos estão na ordem correta. O terceiro elemento é então comparado com o primeiro e, se for maior que o primeiro, é comparado com o segundo elemento; ele é colocado na posição correta entre os primeiros três elementos.

Geralmente, no j-ésimo passo do insertion sort, o j-ésimo elemento da lista é colocado na posição correta na lista que ordena os $j-1$ elementos anteriores. Para inserir o j-ésimo elemento na lista, uma técnica de busca linear é usada (veja o Exercício 43); o j-ésimo elemento é sucessivamente comparado com os $j-1$ elementos já ordenados do começo da lista até que o primeiro elemento que não for menor que este elemento seja encontrado ou até que ele seja comparado com todos os $j-1$ elementos; o j-ésimo elemento é inserido na posição correta, na qual os primeiros j elementos tenham sido ordenados. O algoritmo continua até o último elemento ser colocado na posição correta relativa à lista já ordenada dos primeiros $n-1$ elementos. O insertion sort é descrito em pseudocódigo no Algoritmo 5.

EXEMPLO 5 Use o insertion sort para colocar os elementos da lista 3, 2, 4, 1, 5 em ordem crescente.

Solução: O insertion sort compara primeiro 2 com 3. Como $3 > 2$, ele coloca 2 na primeira posição, produzindo a lista 2, 3, 4, 1, 5 (a parte ordenada da lista está em cinza). Neste ponto, 2 e 3 estão na posição correta. Depois, é inserido o terceiro elemento, 4, na parte já ordenada da lista, fazendo as comparações $4 > 2$ e $4 > 3$. Como $4 > 3$, 4 é colocado na terceira posição. Aqui, a lista é 2, 3, 4, 1, 5 e sabemos que o ordenamento dos três primeiros elementos está correto. Depois, encontramos o lugar certo para o quarto elemento, 1, entre os elementos já ordenados, 2, 3, 4. Como $1 < 2$, obtemos a lista 1, 2, 3, 4, 5. Por fim, inserimos 5 na posição correta pela comparação sucessiva deste elemento com 1, 2, 3 e 4. Como $5 > 4$, ele é posto no final da lista, produzindo a ordem correta de toda a lista. ◀

ALGORITMO 5 O Insertion Sort.

procedure *insertion sort*($a_1, a_2, \ldots, a_n$: real {números reais com $n \geq 2$})
for $j := 2$ **to** n
begin
  $i := 1$
  while $a_j > a_i$
    $i := i + 1$
  $m := a_j$
  for $k := 0$ **to** $j - i - 1$
    $a_{j-k} := a_{j-k-1}$
  $a_i := m$
end {$a_1, a_2, \ldots, a_n$ foram ordenados}

Algoritmo Voraz

Muitos algoritmos que estudaremos neste livro são criados para resolver **problemas de otimização.** O objetivo desses problemas é encontrar uma solução para o problema dado que minimize ou maximize o valor de algum parâmetro. Problemas de otimização estudados mais adiante neste texto incluem encontrar uma rota entre duas cidades que tenha a menor distância, determinar uma maneira de codificar mensagens usando o menor número de bits possível e encontrar um conjunto de ligações de fibras entre nós de redes de transmissão usando a menor quantidade de fibra.

Surpreendentemente, uma das abordagens mais simples geralmente leva à solução de um problema de otimização. Esta abordagem seleciona a melhor escolha para cada passo, em vez de considerar todas as seqüências de passos que podem levar a uma solução ótima. Os algoritmos que fazem o que parece ser a "melhor" escolha em cada passo são chamados de **algoritmos vorazes.** Uma vez que sabemos que um algoritmo voraz encontra uma solução possível, precisamos determinar se foi encontrada uma solução ótima. Para fazer isto, nós demonstramos que a solução é ótima ou mostramos que há um contra-exemplo onde o algoritmo produz uma solução que não é ótima. Para tornar esses conceitos mais concretos, vamos considerar um algoritmo que faz trocos usando moedas.

Links

EXEMPLO 6 Considere o problema de fazer trocos com n centavos com moedas de 25, 10, 5 e 1 centavo e usar o menor número de moedas. Podemos construir um algoritmo voraz para organizar o troco de n centavos fazendo uma escolha ótima em cada passo; ou seja, em cada passo nós escolhemos a moeda com a maior denominação possível para adicionar na pilha de trocos sem exceder n centavos. Por exemplo, para organizar 67 centavos, primeiro selecionamos uma moeda de 25 centavos (faltando 42 centavos). Depois, selecionamos uma segunda moeda de 25 centavos (faltando 17 centavos), seguida de uma moeda de 10 centavos (faltando 7 centavos), depois uma moeda de 5 centavos (faltando 2 centavos), seguida de uma moeda de 1 centavo (faltando 1 centavo) e, por fim, uma moeda de 1 centavo. ◀

Demo

Apresentaremos um algoritmo voraz para organizar trocos, usando qualquer conjunto de moedas, como o Algoritmo 6.

ALGORITMO 6 Algoritmo Voraz para Organizar Trocos.

procedure *troco*($c_1, c_2, \ldots, c_r$: valores das moedas, em que $c_1 > c_2 > \cdots > c_r$; n: inteiro)
for $i := 1$ **to** r
  while $n \geq c_i$
  begin
    adicione uma moeda com valor c_i ao troco
    $n := n - c_i$
  end

Descrevemos um algoritmo voraz para organizar trocos usando moedas de 25, 10, 5 e 1 centavo. Mostraremos que este algoritmo leva a uma solução ótima no sentido que usa o menor número de moedas possível. Antes de desenvolver nossa demonstração, mostraremos que há conjuntos de moedas para as quais o algoritmo voraz (Algoritmo 6) não produz necessariamente trocos usando a menor quantidade possível de moedas. Por exemplo, se temos apenas moedas de 25, 10 e 1 centavo (e não temos as de 5 centavos) para serem usadas, o algoritmo voraz organizaria trocos para 30 centavos usando seis moedas — uma moeda de 25 e cinco moedas de 1 centavo — em vez de usarmos três moedas de 10 centavos.

LEMA 1 Se n for um número inteiro positivo, então n centavos no troco usando moedas de 25, 10, 5 e 1 centavo, com o menor número de moedas possível, terá no máximo duas moedas de 10 centavos, uma moeda de 5 centavos, quatro moedas de 1 centavo, e não pode ter duas moedas de 10 e uma de 5 centavos. A quantia de trocos em moedas de 10, 5 e 1 centavo não pode exceder 24 centavos.

Demonstração: Usamos uma demonstração por contradição. Mostraremos que se tivermos mais que o número específico de moedas de cada tipo, poderemos reorganizá-las usando o menor número de moedas com o mesmo valor. Notamos que se tivermos três moedas de 10 centavos, poderemos substituí-las por uma moeda de 25 centavos e uma de 5 com o mesmo valor; se tivermos duas moedas de 5 centavos, teremos o mesmo valor com uma moeda de 10 centavos se tivermos cinco moedas de 1 centavo, podemos substituí-las por uma de 5 centavos; e se tivermos duas moedas de 10 centavos e uma de 5, teremos o mesmo valor com uma moeda de 25 centavos. Como podemos ter no máximo duas moedas de 10, uma moeda de 5 e quatro moedas de 1 centavo, mas não podemos ter duas moedas de 10 e uma de 5, temos que 24 centavos é o máximo de troco que podemos ter com moedas de 10, 5 e 1 centavo quando organizamos trocos usando o menor número de moedas para n centavos. ◁

TEOREMA 1 O algoritmo voraz (Algoritmo 6) produz trocos usando o menor número de moedas possível.

Demonstração: Usaremos uma demonstração por contradição. Suponha que há um número inteiro positivo n tal que há uma maneira de organizar trocos para n centavos usando moedas de 25, 10, 5 e 1 centavo que usa um número menor de moedas que o algoritmo voraz encontra. Primeiro notamos que q', o número de moedas de 25 centavos usadas na maneira ótima de organizar o troco para n centavos, deve ser o mesmo que q, o número de moedas de 25 centavos usadas pelo algoritmo voraz. Para mostrar isso, primeiro note que o algoritmo voraz usa o maior número de moedas de 25 centavos possível, assim $q' \leq q$. Entretanto, é também o caso de que q' não pode ser menor que q. Se for, poderíamos precisar organizar pelo menos 25 centavos em moedas de 10, 5 e 1 centavo nessa maneira ótima. Contudo. isso é impossível pelo Lema 1.

Como deve haver o mesmo número de moedas de 25 centavos nas duas maneiras de organizar o troco, o valor das moedas de 10, 5 e 1 centavo deve ser o mesmo, e essas moedas não somam mais que 24 centavos. Deve haver o mesmo número de moedas de 10 centavos, porque o algoritmo voraz usa o maior número possível de moedas de 10 centavos e, pelo Lema 1, quando o troco é feito usando-se o menor número de moedas possível, no máximo uma moeda de 5 centavos e no máximo quatro moedas de 1 centavo são usadas, para que seja possível usar o maior número possível de moedas de 10 centavos na maneira ótima. Da mesma forma, temos o mesmo número de moedas de 5 centavos e, por fim, o mesmo número de moedas de 1 centavo. ◁

O Problema da Parada

Vamos descrever agora uma demonstração de um dos mais famosos teoremas em ciência da computação. Mostraremos que há um problema que não pode ser resolvido usando-se qualquer procedimento. Ou seja, mostraremos que há problemas que não podem ser resolvidos. O problema que estudaremos é o **problema da parada**. Ele pergunta se há um procedimento que faça isto: Ele toma como uma entrada um programa de computação e determina se ele irá parar eventualmente quando essa entrada estiver sendo executada. Seria conveniente ter tal procedimento, se ele existisse. Certamente, seríamos capazes de testar, ao escrever, se um programa entra em um laço infinito e acabaríamos com tais defeitos dos programas. Entretanto, em 1936, Alan Turing mostrou que não existe procedimento que cumpra tal objetivo (veja sua biografia na Seção 12.4).

Antes de apresentarmos uma demonstração de que o problema da parada não pode ser resolvido, note que não podemos simplesmente executar um programa e observar o que ele faz para determinar se finalizará quando executa com a entrada dada. Se o programa parar, nós temos nossa resposta, mas se ele continuar rodando ainda depois de fixado qualquer espaço de tempo, não sabemos se nunca irá parar ou se apenas não esperou o tempo suficiente para finalizar. Afinal, não é difícil criar um programa que irá parar apenas depois de bilhões de anos.

Vamos descrever a demonstração de Turing de que o problema da parada não pode ser resolvido; é uma demonstração por contradição. (O leitor deve notar que nossa demonstração não é completamente rigorosa, pois não definimos explicitamente qual é o procedimento. Para corrigir isso, o conceito de máquina de Turing é necessário. Este conceito é apresentado na Seção 12.5.)

Demonstração: Suponha que haja uma solução para o problema da parada, um procedimento chamado de $H(P, I)$. Esse procedimento tem duas entradas, uma, um programa P e a outra, I, uma

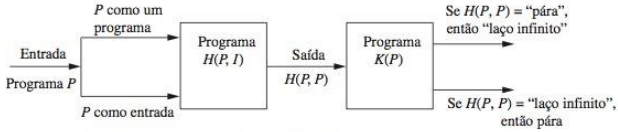


FIGURA 2 Mostra que o Problema da Parada não tem Solução.

entrada para o programa P. $H(P, I)$ constrói a seqüência "pára" como saída se H determinar que P deve parar quando for dada I como entrada. Caso contrário, $H(P, I)$ constrói a seqüência "laço infinito" como saída. Vamos derivar uma contradição.

Quando um procedimento é codificado, é expresso como uma seqüência de caracteres; esta seqüência pode ser interpretada como uma seqüência de bits. Isto significa que um programa pode ser usado como dado. Então, ele pode ser pensado como uma entrada para outro programa, ou mesmo para ele próprio. Assim, H pode ser considerado como um programa P como suas entradas, que são um programa e a entrada para este programa. H deveria ser capaz de determinar se P irá parar quando for dada uma cópia dele mesmo como entrada.

Para mostrar que não existe o procedimento H que resolva o problema da parada, construímos um procedimento simples $K(P)$, que funciona fazendo uso da saída $H(P, P)$. Se a saída de $H(P, P)$ é um "laço infinito", o que significa que P entra em laço eternamente quando é dada uma cópia dele mesmo como entrada, então $K(P)$ pára. Se a saída de $H(P, P)$ é "pára", o que significa que P pára quando é dada uma cópia dele mesmo como entrada, então $K(P)$ entra em laço infinito. Ou seja, $K(P)$ faz o oposto do que especifica a saída de $H(P, P)$. (Veja a Figura 2.)

Agora suponha que nós fornecemos K como entrada para K. Notamos que se a saída de $H(K, K)$ é o "laço infinito", então, pela definição de K, vemos que $K(K)$ pára. Por outro lado, se a saída de $H(K, K)$ é "pára", então, pela definição de K, vemos que $K(K)$ continua em laço infinito, ao contrário do que H nos diz. Nos dois casos temos uma contradição.

Então, H não pode dar as respostas corretas sempre. Conseqüentemente, não há procedimento que resolva o problema da parada. ◁

Exercícios

1. Liste todos os passos usados pelo Algoritmo 1 para encontrar o valor máximo na lista 1, 8, 12, 9, 11, 2, 14, 5, 10, 4.
2. Determine quais características de um algoritmo há nos procedimentos a seguir e quais características estão faltando.
   a) **procedure** *dobro*(n: inteiro)
      while $n > 0$
         $n := 2n$
   b) **procedure** *divisão*(n: inteiro)
      while $n \geq 0$
      begin
         $m := 1/n$
         $n := n - 1$
      end
   c) **procedure** *soma*(n: inteiro)
      $soma := 0$
      while $i < 10$
         $soma := soma + i$
   d) **procedure** *escolha*(a, b: inteiro)
      $x :=$ a ou b
3. Construa um algoritmo que encontre a soma de todos os números inteiros de uma lista.
4. Descreva um algoritmo que considere como entrada uma lista de n números inteiros e produza como saída a maior diferença obtida pela subtração de um número inteiro da lista e seu subseqüente.
5. Descreva um algoritmo que considere como entrada uma lista de n números inteiros em ordem não crescente e produza a lista de todos os valores que aparecem mais de uma vez.
6. Descreva um algoritmo que considere como entrada uma lista de n números inteiros e encontre o número de inteiros negativos da lista.
7. Descreva um algoritmo que considere como entrada uma lista de n números inteiros e encontre a localização do último número inteiro par na lista ou retorne 0 se não houver números inteiros pares na lista.
8. Descreva um algoritmo que considere como entrada uma lista de n números inteiros distintos e encontre a localização do maior número inteiro par na lista ou retorne 0 se não houver números inteiros pares na lista.

9. Um **palíndromo** é uma seqüência que pode ser lida de trás para frente e de frente para trás. Descreva um algoritmo para determinar se uma seqüência de n caracteres é um palíndromo.

10. Construa um algoritmo para computar x^n, em que x é um número real e n é um número inteiro. [*Dica:* Primeiro forneça um procedimento para computar x^n quando n for um número não negativo por sucessivas multiplicações por x, começando com 1. Então, estenda este procedimento e use o fato de que $x^{-n} = 1/x^n$ para computar x^n quando n for negativo.]

11. Descreva um algoritmo que troque os valores das variáveis x e y, usando apenas tarefas. Qual é o menor número de proposições necessárias para fazer isto?

12. Descreva um algoritmo que use apenas proposições que substitua o trio (x, y, z) em (y, z, x). Qual é o menor número de proposições necessárias?

13. Liste todos os passos usados para procurar o número 9 na seqüência 1, 3, 4, 5, 6, 8, 9, 11 usando

a) uma busca linear. **b)** uma busca binária.

14. Liste todos os passos usados para encontrar o número 7 na seqüência dada no Exercício 13.

15. Descreva um algoritmo que insere um número inteiro x na posição apropriada na lista $a_1, a_2, \ldots, a_n$ de números inteiros que está em ordem crescente.

16. Descreva um algoritmo que possa encontrar o menor número inteiro em uma seqüência finita de números naturais.

17. Descreva um algoritmo que localize a primeira ocorrência do maior elemento em uma lista finita de números inteiros, na qual os inteiros não são necessariamente distintos.

18. Descreva um algoritmo que localize a última ocorrência do menor elemento em uma lista finita de números inteiros, na qual os inteiros não são necessariamente distintos.

19. Descreva um algoritmo que produza os valores máximo, mediano, mínimo e a média de um conjunto de três números inteiros. (O valor **mediano** de um conjunto de números inteiros é o elemento central da lista quando esses inteiros estão em ordem crescente de tamanho. A **média** de um conjunto de números inteiros é a soma dos inteiros dividida pelo número de inteiros no conjunto.)

20. Descreva um algoritmo que possa encontrar o maior e o menor número inteiro em uma seqüência finita de inteiros.

21. Descreva um algoritmo que coloca os primeiros três termos de uma seqüência de números inteiros de comprimento arbitrário em ordem crescente.

22. Descreva um algoritmo que possa encontrar a maior palavra em uma sentença em português (uma sentença é uma seqüência de símbolos, ou uma letra ou um espaço, que pode ser quebrada alternando-se palavras e espaços).

23. Descreva um algoritmo que determine se uma função de um conjunto finito de números inteiros para outro conjunto finito de números inteiros é sobrejetiva.

24. Descreva um algoritmo que determina se uma função de um conjunto finito para outro conjunto finito é injetora.

25. Descreva um algoritmo que irá contar o número de 1s em uma cadeia de bits examinando cada bit da cadeia para determinar se é um bit 1.

26. Mude o Algoritmo 3 para que o procedimento de busca binária compare x com a_m em cada etapa do algoritmo, com o algoritmo terminando se $x = a_m$. Qual a vantagem desta versão do algoritmo?

27. O **algoritmo de busca ternária** localiza um elemento em uma lista de números inteiros crescentes por sucessivas divisões da lista em três sublistas de tamanho igual (ou o mais próximo possível de serem iguais) e restringe a busca na sublista apropriada. Especifique os passos deste algoritmo.

28. Especifique os passos de um algoritmo que localiza um elemento em uma lista de números inteiros crescentes por sucessivas divisões da lista em quatro sublistas de tamanho igual (ou o mais próximo possível de serem iguais) e restringe a busca na sublista apropriada.

29. Uma **moda** de uma lista de números inteiros é um elemento que ocorre ao menos tão freqüentemente quanto os outros elementos. Construa um algoritmo que encontre a moda de uma lista de números inteiros não decrescente.

30. Construa um algoritmo que possa encontrar todas as modas (definido no Exercício 29) de uma lista de números inteiros não decrescente.

31. Construa um algoritmo que possa encontrar o primeiro termo de uma seqüência de números inteiros que seja igual a algum termo anterior da seqüência.

32. Construa um algoritmo que possa encontrar todos os termos de uma seqüência finita de números inteiros que sejam maiores que a soma de todos os termos anteriores da seqüência.

33. Construa um algoritmo que possa encontrar o primeiro termo de uma seqüência de números inteiros positivos que seja menor que o termo imediatamente anterior da seqüência.

34. Use o bubble sort para ordenar 6, 2, 3, 1, 5, 4, mostrando a lista obtida em cada passo.

35. Use o bubble sort para ordenar 3, 1, 5, 7, 4, mostrando a lista obtida em cada passo.

36. Use o bubble sort para ordenar d, f, k, m, a, b, mostrando a lista obtida em cada passo.

***37.** Adapte o algoritmo bubble sort para que ele pare quando não houver mais trocas. Expresse esta versão mais eficiente do algoritmo em pseudocódigo.

38. Use o insertion sort para ordenar a lista do Exercício 34, mostrando as listas obtidas em cada passo.

39. Use o insertion sort para ordenar a lista do Exercício 35, mostrando as listas obtidas em cada passo.

40. Use o insertion sort para ordenar a lista do Exercício 36, mostrando as listas obtidas em cada passo.

O **selection sort** começa encontrando o menor elemento da lista. Este elemento é movido para frente. Então, o menor elemento entre os restantes é encontrado e colocado na segunda posição. Este procedimento é repetido até que a lista toda seja ordenada.

41. Ordene as listas abaixo usando o selection sort.

a) 3, 5, 4, 1, 2 **b)** 5, 4, 3, 2, 1

c) 1, 2, 3, 4, 5

42. Escreva o algoritmo de selection sort em pseudocódigo.

43. Descreva um algoritmo baseado na busca linear para determinar a posição correta na qual é inserido um novo elemento na lista já ordenada.

44. Descreva um algoritmo baseado na busca binária para determinar a posição correta na qual é inserido um novo elemento em uma lista já ordenada.

45. Quantas comparações um insertion sort usa para ordenar a lista $1, 2, \ldots, n$?

46. Quantas comparações um insertion sort usa para ordenar a lista $n, n-1, \ldots, 2, 1$?

O **insertion sort binário** é uma variação do insertion sort que usa uma técnica de busca binária (veja o Exercício 44) em vez de uma técnica de busca linear para inserir o i-ésimo elemento no lugar correto entre os elementos previamente ordenados.

47. Mostre todos os passos usados pelo insertion sort binário para ordenar a lista 3, 2, 4, 5, 1, 6.

48. Verifique as diferenças entre o número de comparações usadas pelo insertion sort e pelo insertion sort binário para ordenar a lista 7, 4, 3, 8, 1, 5, 4, 2.

***49.** Expresse o insertion sort binário em pseudocódigo.

50. **a)** Construa uma variação do insertion sort que use uma técnica de busca linear para inserir o j-ésimo elemento no lugar correto comparando-o primeiro com o $(j-1)$-ésimo elemento, e depois com o $(j-2)$-ésimo elemento se necessário, e assim por diante.

b) Use seu algoritmo para ordenar 3, 2, 4, 5, 1, 6.

c) Responda ao Exercício 45 usando esse algoritmo.

d) Responda ao Exercício 46 usando esse algoritmo.

51. Quando uma lista de elementos está próxima da ordem correta, é melhor usar o insertion sort ou sua variação descrita no Exercício 50?

52. Use o algoritmo voraz para organizar os seguintes trocos usando moedas de 25, 10, 5 e 1 centavo:

a) 87 centavos. **b)** 49 centavos.

c) 99 centavos. **d)** 33 centavos.

53. Use o algoritmo voraz para organizar os seguintes trocos usando moedas de 25, 10, 5 e 1 centavo:

a) 51 centavos. **b)** 69 centavos.

c) 76 centavos. **d)** 60 centavos.

54. Use o algoritmo voraz para organizar os seguintes trocos usando moedas de 25, 10 e 1 centavo (mas não as de 5 centavos) para cada quantia dada no Exercício 52. Para cada uma dessas quantias, o algoritmo voraz usa o menor número de moedas possível?

55. Use o algoritmo voraz para organizar os seguintes trocos usando moedas de 25, 10 e 1 centavo (mas não as de 5 centavos) para cada quantia dada no Exercício 53. Para cada uma dessas quantias, o algoritmo voraz usa o menor número de moedas possível?

56. Mostre que se houvesse uma moeda que valesse 12 centavos, o algoritmo voraz, ao usar moedas de 25, 12, 10, 5 e 1 centavo, não produziria sempre o troco usando o menor número de moedas possível.

57. **a)** Descreva um algoritmo voraz que resolva o problema de organizar um subconjunto de uma coleção de palestras apresentadas em um auditório selecionando cada passo da palestra proposta que tem o final mais curto entre todos aqueles que não entram em conflito com as palestras já organizadas. (Na Seção 4.1 mostraremos que este algoritmo voraz sempre produz um horário que inclui o maior número de palestras possível.)

b) Use o seu algoritmo do item (a) para organizar o maior número possível de palestras em um auditório, a partir de um conjunto de palestras apresentadas, se os horários das palestras são 9h às 9h45; 9h30 às 10h; 9h50 às 10h15; 10h às 10h30; 10h10 às 10h25; 10h30 às 10h55; 10h15 às 10h45; 10h30 às 11h; 10h45 às 11h30; 10h55 às 11h25; 11h às 11h15.

Suponha que tenhamos s homens $m_1, m_2, \ldots, m_s$ e s mulheres $w_1, w_2, \ldots, w_s$. Queremos que cada pessoa seja colocada com outra do sexo oposto. Além disso, suponha que cada pessoa escolha, em ordem de preferência, e sem dúvidas, as pessoas do sexo oposto. Dizemos que um conjunto de pessoas de sexos opostos para formar casal é **estável** se não pudermos encontrar um homem m e uma mulher w que não são agrupados em uma situação em que m prefere w como sua parceira determinada e w prefere m como seu parceiro.

58. Suponha que tenhamos três homens m_1, m_2 e m_3 e três mulheres w_1, w_2 e w_3. Além disso, suponha que as listas de preferência dos homens para as três mulheres, da maior para a menor, são m_1: w_3, w_1, w_2; m_2: w_1, w_2, w_3; m_3: w_2, w_3, w_1; e as listas de preferências das mulheres para os três homens, do maior para o menor, são w_1: m_1, m_2, m_3; w_2: m_2, m_1, m_3; w_3: m_3, m_2, m_1. Para cada um dos seis agrupamentos possíveis de homens e mulheres para formar três casais, determine se esses agrupamentos são estáveis.

***59.** O **algoritmo de aceitação** pode ser usado para construir um agrupamento de homens e mulheres. Neste algoritmo, os membros de um sexo são **os pretendentes** e os membros do outro sexo, **os pretendidos**. O algoritmo usa uma série de rodadas; em cada rodada, todo pretendente cuja proposta foi rejeitada na rodada anterior faz uma proposta ao seu maior pretendido que ainda não tenha rejeitado uma proposta sua. O pretendido rejeita todas as propostas, exceto aquela do pretendente que ele tem como maior entre todos os outros que fizeram propostas nesta ou nas rodadas anteriores. A proposta deste maior pretendente permanece pendente e é rejeitada em uma próxima rodada se houver mais um pretendente propondo naquela rodada. As séries de rodadas terminam quando todo pretendente tiver uma proposta pendente. Todas as propostas pendentes são então aceitas.

a) Escreva o algoritmo de aceitação em pseudocódigo.

b) Mostre que o algoritmo de aceitação termina.

c) Mostre que a aceitação sempre termina com um agrupamento estável.

60. Mostre que o problema de determinar se um programa com uma dada entrada sempre imprime o dígito 1 é insolúvel.

61. Mostre que o problema a seguir pode ser resolvido. Dados dois programas com suas entradas e o conhecimento de que uma delas pára, determine qual pára.

62. Mostre que o problema de decidir se um programa específico com uma entrada específica pára pode ser resolvido.

3.2 O Crescimento de Funções

Introdução

Na Seção 3.1, discutimos o conceito de algoritmo. Introduzimos os algoritmos que resolvem vários problemas, incluindo a busca por um elemento em uma lista e a ordenação de uma lista. Na Seção 3.3, estudaremos o número de operações usadas por esses algoritmos. Vamos estimar especialmente o número de comparações usadas por algoritmos de busca linear e binária para encontrar um elemento em uma seqüência de n elementos. Vamos também fazer uma estimativa do número de comparações usadas pelo bubble sort e pelo insertion sort para ordenar uma lista de n elementos. O tempo necessário para resolver um problema depende não só do número de operações que ele usa, mas também do hardware e do software usados para abrir o programa que implementa o algoritmo. Entretanto, quando mudamos o hardware e o software usados para implementar um algoritmo, podemos nos aproximar do tempo necessário para resolver um problema de tamanho n pela multiplicação dos tempos anteriores exigidos por uma constante. Por exemplo, em um supercomputador podemos resolver um problema de tamanho n um milhão de vezes mais rápido do que em um PC. Entretanto, este fator de um milhão não depende de n (exceto talvez em termos de ordem menor). Uma das vantagens de usar a **notação O grande,** que introduziremos nesta seção, é que podemos estimar o crescimento de uma função sem nos preocuparmos com as constantes multiplicadoras ou termos de ordens menores. Isto significa que, usando a notação O grande, não precisamos nos preocupar com o hardware e o software usados para implementar um algoritmo. Além disso, com essa notação, podemos assumir que as operações diferentes usadas em um algoritmo levam o mesmo tempo, o que simplifica consideravelmente a análise.

A notação O grande é muito usada para estimar o número de operações que um algoritmo utiliza à medida que sua entrada aumenta. Com a ajuda dessa notação, podemos determinar se é prático usar um determinado algoritmo para resolver um problema quando o tamanho de sua entrada aumenta. Além disso, usando a notação O grande, podemos comparar dois algoritmos para determinar qual é o mais eficiente, dependendo do tamanho da entrada. Por exemplo, se tivermos dois algoritmos para resolver um problema, um usando $100n^2 + 17n + 4$ operações e o outro, n^3 operações, a notação O grande pode nos ajudar a ver se o primeiro algoritmo usa menos operações quando n é grande, mesmo se ele usar mais operações para valores menores de n, tal como $n = 10$.

Esta seção introduz a notação O grande e as notações relacionadas Ômega grande e Teta grande. Explicaremos como as estimativas O grande, Ômega grande e Teta grande são construídas e estabelecidas para algumas funções importantes que são usadas na análise de algoritmos.

Notação O grande

O crescimento de funções é geralmente descrito usando-se uma notação especial. A Definição 1 descreve esta notação.

DEFINIÇÃO 1 Sejam f e g as funções do conjunto de números inteiros ou dos números reais para o conjunto dos números reais. Dizemos que $f(x)$ é $O(g(x))$ se houver constantes C e k, tal que

$$|f(x)| \leq C\,|g(x)|$$

sempre que $x > k$. [Isto é lido como "$f(x)$ é O grande de $g(x)$."]

As constantes C e k na definição da notação O grande são chamadas de **parâmetros** da relação $f(x)$ é $O(g(x))$. Para estabelecer que $f(x)$ é $O(g(x))$ precisamos apenas de um par de parâmetros para esta relação. Ou seja, para mostrar que $f(x)$ é $O(g(x))$, precisamos encontrar *um* par de constantes C e k, os parâmetros, tal que $|f(x)| \leq C|g(x)|$ sempre que $x > k$.

Auto-avaliação

Note que quando há um par de parâmetros para a relação $f(x)$ é $O(g(x))$, há *infinitos* pares de parâmetros. Para verificar isso, note que se C e k formam um par de parâmetros, então qualquer par C' e k', em que $C < C'$ e $k < k'$, é também um par de parâmetros, porque $|f(x)| \leq C|g(x)| \leq C'|g(x)|$ sempre que $x > k' > k$.

Links

Um método útil para encontrar um par de parâmetros é primeiro selecionar um valor de k para o qual o tamanho de $|f(x)|$ pode ser rapidamente estimado quando $x > k$ e ver se podemos usar essa estimativa para encontrar um valor de C para o qual $|f(x)| < C|g(x)|$ para $x > k$. Esta aproximação é ilustrada no Exemplo 1.

EXEMPLO 1 Mostre que $f(x) = x^2 + 2x + 1$ é $O(x^2)$.

Exemplos Extras

Solução: Observamos que podemos estimar rapidamente o tamanho de $f(x)$ quando $x > 1$, pois $x < x^2$ e $1 < x^2$ quando $x > 1$. Temos que

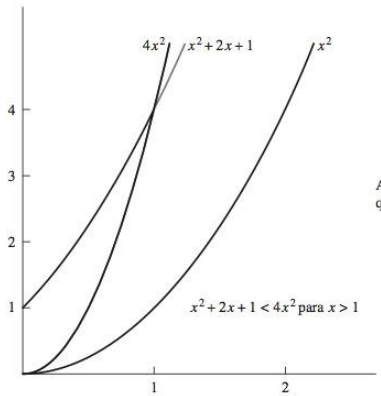
$$0 \leq x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2 = 4x^2$$

sempre que $x > 1$, como mostrado na Figura 1. Conseqüentemente, podemos ter $C = 4$ e $k = 1$ como parâmetros para mostrar que $f(x)$ é $O(x^2)$. Ou seja, $f(x) = x^2 + 2x + 1 < 4x^2$ sempre que $x > 1$. (Note que não é necessário usar valores absolutos aqui porque todas as funções nesta igualdade são positivas quando x é positivo.)

Alternativamente, podemos estimar o tamanho de $f(x)$ quando $x > 2$. Quando $x > 2$, temos $2x \leq x^2$ e $1 \leq x^2$. Conseqüentemente, se $x > 2$, temos

$$0 \leq x^2 + 2x + 1 \leq x^2 + x^2 + x^2 = 3x^2.$$

Segue-se que $C = 3$ e $k = 2$ são também parâmetros da relação em que $f(x)$ é $O(x^2)$.



A parte do gráfico de $f(x) = x^2 + 2x + 1$ que satisfaz $f(x) < 4x^2$ está em cinza.

FIGURA 1 **A Função $x^2 + 2x + 1$ é $O(x^2)$.**

Observe que na relação "$f(x)$ é $O(x^2)$," x^2 pode ser trocado por qualquer função com valores maiores que x^2. Por exemplo, $f(x)$ é $O(x^3)$, $f(x)$ é $O(x^2 + x + 7)$, e assim por diante.

É verdade também que x^2 é $O(x^2 + 2x + 1)$, porque $x^2 < x^2 + 2x + 1$ sempre que $x > 1$. Isto significa que $C = 1$ e $k = 1$ são parâmetros da relação em que x^2 é $O(x^2 + 2x + 1)$. ◀

Note que no Exemplo 1 temos duas funções, $f(x) = x^2 + 2x + 1$ e $g(x) = x^2$, tal que $f(x)$ é $O(g(x))$ e $g(x)$ é $O(f(x))$ — o último fato vem da inequação $x^2 \leq x^2 + 2x + 1$, que é garantida para todos os números reais não negativos x. Dizemos que duas funções $f(x)$ e $g(x)$ que satisfazem a essas relações de O grande são de **mesma ordem.** Retomaremos esta noção mais à frente nesta seção.

Lembre-se: O fato de dizermos que $f(x)$ é $O(g(x))$ é às vezes escrito como $f(x) = O(g(x))$. Entretanto, o sinal de igual *não* representa nesta notação uma igualdade genuína. Ao contrário, esta notação nos diz que uma inequação é sustentada pela relação entre valores das funções f e g para os números suficientemente grandes no domínio dessas funções. Entretanto, é aceitável escrever $f(x) \in O(g(x))$ porque $O(g(x))$ representa o conjunto de funções que são $O(g(x))$.

A notação O grande tem sido usada por matemáticos há mais de um século. Em ciência da computação, ela é muito usada na análise de algoritmos, como veremos na Seção 3.3. O matemático alemão Paul Bachmann foi o primeiro a introduzir essa notação em 1892, em um importante livro sobre teoria dos números. O símbolo O grande é às vezes chamado de **símbolo de Landau**, em homenagem ao matemático alemão Edmund Landau, que usou essa notação em seu trabalho. O uso da notação O grande em ciência da computação foi popularizado por Donald Knuth, que também introduziu as notações Ω grande e Θ grande, que serão definidas ao longo desta seção.

Quando $f(x)$ é $O(g(x))$ e $h(x)$ é uma função que tem os valores absolutos maiores que $g(x)$ para os maiores valores suficientes de x, temos que $f(x)$ é $O(h(x))$. Em outras palavras, a função $g(x)$ na relação em que $f(x)$ é $O(g(x))$ pode ser substituída por uma função com valores absolutos maiores. Para verificar isso, note que se

$$|f(x)| \leq C\,|g(x)| \qquad \text{se } x > k,$$

e se $|h(x)| > |g(x)|$ para todo $x > k$, então

$$|f(x)| \leq C\,|h(x)| \qquad \text{se } x > k.$$

Assim, $f(x)$ é $O(h(x))$.

Quando a notação O grande é usada, a função g na relação em que $f(x)$ é $O(g(x))$ é escolhida como a menor possível (às vezes de um conjunto de funções referenciais, tal como funções na forma x^n, em que n é um número inteiro positivo).

Links

PAUL GUSTAV HEINRICH BACHMANN (1837–1920) Paul Bachmann, filho de um pastor luterano, compartilhou o estilo de vida devoto de seu pai e também seu amor pela música. Seu talento matemático foi descoberto por um de seus professores, mesmo ele tendo dificuldades com seus primeiros estudos matemáticos. Depois de se recuperar da tuberculose na Suíça, Bachmann estudou matemática, primeiro na Universidade de Berlim e depois na de Göttingen, onde freqüentava as aulas do famoso teórico dos números Dirichlet. Recebeu seu título de doutorado sob a orientação do teórico alemão Kummer em 1862; sua tese era sobre a teoria dos grupos. Bachmann foi professor em Breslau e depois em Münster. Depois de sua aposentadoria, ele continuou com seus escritos matemáticos e a tocar piano e trabalhou como crítico de música para jornais. Os trabalhos matemáticos de Bachmann incluem uma pesquisa de cinco volumes sobre resultados e métodos em teoria dos números, um trabalho de dois volumes sobre a teoria elementar dos números, outro sobre números irracionais e outro sobre a famosa conjectura conhecida como Último Teorema de Fermat. Ele introduziu a notação O grande em seu livro de 1892, *Analytische Zahlentheorie.*

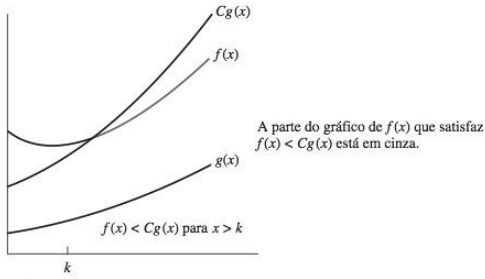


FIGURA 2 **A Função $f(x)$ é $O(g(x))$.**

Nas próximas discussões, quase sempre vamos lidar com funções que apresentam apenas valores positivos. Todas as referências aos valores absolutos podem ser deixadas de lado ao trabalhar com as estimativas O grande para essas funções. A Figura 2 ilustra a relação em que $f(x)$ é $O(g(x))$.

O Exemplo 2 mostra como a notação O grande é usada para estimar o crescimento de funções.

EXEMPLO 2 Mostre que $7x^2$ é $O(x^3)$.

Solução: Note que quando $x > 7$, temos $7x^2 < x^3$. (Podemos obter esta inequação multiplicando os dois lados de $x > 7$ por x^2.) Conseqüentemente, podemos ter que $C = 1$ e $k = 7$ como parâmetros para estabelecer a relação em que $7x^2$ é $O(x^3)$. Alternativamente, quando $x > 1$, temos que $7x^2 < 7x^3$, assim, $C = 7$ e $k = 1$ são também parâmetros para a relação em que $7x^2$ é $O(x^3)$. ◀

O Exemplo 3 mostra como verificar que a relação O grande não se mantém.

EXEMPLO 3 Mostre que n^2 não é $O(n)$.

Solução: Para mostrar que n^2 não é $O(n)$, devemos mostrar que não existe nenhum par de constantes C e k, tal que $n^2 \leq Cn$ sempre que $n > k$. Para ver que não há tais constantes, observe que quando $n > 0$, podemos dividir os dois lados da inequação $n^2 \leq Cn$ por n para obter a inequação equivalente $n \leq C$. Vemos agora que não importa qual o valor de C e k, a inequação $n \leq C$ não pode ser mantida para todo n com $n > k$. Em particular, uma vez que consideremos um valor de k, vemos que quando n é maior que o valor máximo de k e C, não é verdade que $n \leq C$, mesmo se $n > k$. ◀

Links

EDMUND LANDAU (1877–1938) Edmund Landau, filho de um ginecologista de Berlim, freqüentou o colégio e a faculdade nessa mesma cidade. Recebeu seu título de doutorado em 1899, sob a orientação de Frobenius. Landau primeiro ministrou aulas na Universidade de Berlim e depois se mudou para Göttingen, onde trabalhava com dedicação exclusiva até que a força nazista impediu-o de continuar a lecionar. As principais contribuições de Landau para a matemática foram no campo da teoria analítica dos números. Em particular, estabeleceu muitos resultados importantes sobre a distribuição dos números primos. Ele é autor de uma exposição de três volumes sobre a teoria dos números, assim como de outros livros sobre este mesmo tema e sobre análise matemática.

EXEMPLO 4 O Exemplo 2 mostra que $7x^2$ é $O(x^3)$. É verdade também que x^3 é $O(7x^2)$?

Solução: Para determinar se x^3 é $O(7x^2)$, precisamos definir se há constantes C e k, tal que $x^3 \leq C(7x^2)$ sempre que $x > k$. A inequação $x^3 \leq C(7x^2)$ é equivalente à inequação $x \leq 7C$, a qual obtemos pela divisão da inequação original pelo valor positivo x^2. Note que não existe C para o qual $x \leq 7C$ para todo $x > k$, não importa qual o valor de k, pois x pode ser arbitrariamente grande. Temos, portanto, que não existem parâmetros C e k para esta relação proposta. Assim, x^3 *não é* $O(7x^2)$. ◀

Alguns Resultados Importantes de *O* grande

Polinômios geralmente podem ser usados para estimar o crescimento de funções. Em vez de analisar o crescimento de polinômios cada vez que eles aparecem, podemos obter um resultado que sempre será usado para estimar o crescimento de uma função polinomial. O Teorema 1 faz isso. Ele mostra que um dado termo de um polinômio controla seu crescimento ao afirmar que um polinômio de grau n, ou menos, é $O(x^n)$.

TEOREMA 1 Considere $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, em que $a_0, a_1, \ldots, a_{n-1}, a_n$ são números reais. Então, $f(x)$ é $O(x^n)$.

Links

DONALD E. KNUTH (NASCIDO EM 1938) Knuth cresceu em Milwaukee, onde seu pai ensinava ciências contábeis em uma escola luterana e tinha seu próprio negócio de impressão. Ele era um excelente aluno, e ganhou diversos prêmios acadêmicos. Aplicou sua inteligência de maneira não convencional ao vencer um concurso, quando estava na oitava série, por encontrar mais de 4 500 palavras que poderiam ser formadas com as letras de "Ziegler's Giant Bar". Neste evento, ele ganhou um aparelho de televisão para sua escola e um doce para cada um dos colegas de sua sala.

Knuth teve dificuldades ao escolher a física em vez da música como sua graduação no Case Institute of Technology. Ele, então, transferiu-se da física para a matemática e, em 1960, recebeu seu título de bacharel em ciências e, simultaneamente, o título de mestre em ciências por um prêmio especial da faculdade, que considerou seu trabalho excelente. No Case, ele comandou o time de basquete e aplicou seus talentos na construção de uma fórmula para o valor de cada jogador. Esta história foi coberta pela *Newsweek* e por Walter Cronkite, na rede de televisão CBS. Knuth começou seu trabalho de pós-graduação no California Institute of Technology em 1960, e recebeu seu título de Ph.D. em 1963. Durante esse tempo, trabalhou como consultor, escrevendo compiladores para diferentes computadores.

Knuth juntou-se à equipe do California Institute of Technology em 1963, na qual permaneceu até 1968, quando começou a trabalhar como professor da Universidade de Stanford, onde se aposentou como professor emérito em 1992 para se dedicar aos seus escritos. Ele se interessa especialmente por atualizações e por completar novos volumes da sua série *The Art of Computer Programming,* um trabalho que tem uma grande influência no desenvolvimento da ciência da computação, que ele começou a escrever em 1962, com foco nos compiladores. No jargão comum, "Knuth", aludindo a *The Art of Computer Programming,* tornou-se referência às respostas para todas as perguntas sobre tópicos como estruturas de dados e algoritmos.

Knuth é o fundador do estudo moderno da complexidade da computação. Ele fez contribuições fundamentais para o tema compiladores. Sua insatisfação com a tipografia matemática o fez inventar os sistemas amplamente usados atualmente: TeX e Metafont. O sistema TeX tornou-se uma linguagem-padrão para a tipografia computacional. Dois dos muitos prêmios recebidos por Knuth foram o Turing Award, em 1974, e o National Medal of Technology, em 1979, dado a ele pelo Presidente Carter.

Knuth escreveu para vários periódicos científicos sobre ciência da computação e matemática. Entretanto, sua primeira publicação, em 1957, quando ele era um calouro na universidade, foi uma paródia do sistema métrico chamado de "Sistema Potrzebie de Pesos e Medidas," que apareceu na *MAD Magazine* e foi reimpresso muitas vezes. Ele é engajado em sua igreja, assim como foi seu pai. Também é compositor de música para órgão. Knuth acredita que escrever programas para computadores pode ser uma experiência estética, assim como escrever poesia ou compor músicas.

Knuth paga $ 2,56 para a primeira pessoa que encontrar um erro em seus livros e $ 0,32 para sugestões importantes. Se você enviar-lhe uma carta com um erro (será necessário usar os correios, porque ele desistiu de ler e-mails), ele informará se você foi a primeira pessoa a falar sobre aquele erro. Esteja preparado para uma longa espera, porque ele recebe uma grande quantidade de cartas. (O autor deste livro recebeu uma carta alguns anos depois de mandar uma notificação de erro para Knuth, dizendo que aquela notificação tinha chegado alguns meses depois da primeira notificação do erro.)

Demonstração: Usando a inequação triangular (veja o Exercício 5 na Seção 1.7), se $x > 1$, temos

$$\begin{aligned}|f(x)| &= |a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0| \\ &\leq |a_n|x^n + |a_{n-1}|x^{n-1} + \cdots + |a_1|x + |a_0| \\ &= x^n\,(|a_n| + |a_{n-1}|/x + \cdots + |a_1|/x^{n-1} + |a_0|/x^n) \\ &\leq x^n\,(|a_n| + |a_{n-1}| + \cdots + |a_1| + |a_0|).\end{aligned}$$

Isto mostra que

$$|f(x)| \leq Cx^n,$$

em que $C = |a_n| + |a_{n-1}| + \cdots + |a_0|$ sempre que $x > 1$. Assim, os parâmetros $C = |a_n| + |a_{n-1}| + \cdots + |a_0|$ e $k = 1$ mostram que $f(x)$ é $O(x^n)$. ◁

Daremos alguns exemplos que envolvem funções que têm o conjunto dos números inteiros positivos como seu domínio.

EXEMPLO 5 Como a notação O grande pode ser usada para estimar a soma dos primeiros n números inteiros positivos?

Solução: Como cada um dos números inteiros na soma dos primeiros n números inteiros positivos não excede a n, temos que

$$1 + 2 + \cdots + n \leq n + n + \cdots + n = n^2.$$

A partir dessa inequação, temos que $1 + 2 + 3 + \cdots + n$ é $O(n^2)$, tomando $C = 1$ e $k = 1$ como parâmetros. (Neste exemplo, o domínio das funções na relação O grande é o conjunto dos números inteiros positivos.) ◀

No Exemplo 6, a estimativa O grande será desenvolvida para a função fatorial e seu logaritmo. Essas estimativas serão importantes na análise do número de passos usados em procedimentos de ordenação.

EXEMPLO 6 Dê a estimativa O grande para a função fatorial e para o logaritmo dessa função, em que a função fatorial $f(n) = n!$ é definida por

$$n! = 1 \cdot 2 \cdot 3 \cdot \cdots \cdot n$$

sempre que n for um número inteiro positivo e $0! = 1$. Por exemplo,

$$1! = 1, \qquad 2! = 1 \cdot 2 = 2, \qquad 3! = 1 \cdot 2 \cdot 3 = 6, \qquad 4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24.$$

Note que a função $n!$ cresce rapidamente. Por exemplo,

$$20! = 2\,432\,902\,008\,176\,640\,000.$$

Solução: Uma estimativa O grande para $n!$ pode ser obtida notando que cada termo no produto não excede a n. Assim,

$$\begin{aligned}n! &= 1 \cdot 2 \cdot 3 \cdot \cdots \cdot n \\ &\leq n \cdot n \cdot n \cdot \cdots \cdot n \\ &= n^n.\end{aligned}$$

Esta inequação mostra que $n!$ é $O(n^n)$, tomando $C = 1$ e $k = 1$ como parâmetros. Tomando os logaritmos dos dois lados da inequação estabelecida para $n!$, obtemos

$$\log n! \leq \log n^n = n \log n.$$

Isto implica que $\log n!$ é $O(n \log n)$, tomando novamente $C = 1$ e $k = 1$ como parâmetros. ◀

EXEMPLO 7 Na Seção 4.1, mostraremos que $n < 2^n$ sempre que n for um número inteiro positivo. Mostre que a inequação implica que n é $O(2^n)$, e use a inequação para mostrar que $\log n$ é $O(n)$.

Solução: Usando a inequação $n < 2^n$, rapidamente concluímos que n é $O(2^n)$ tomando $k = C = 1$ como parâmetros. Note que, como a função logarítmica é crescente, tomando logaritmos (de base 2) dos dois lados da inequação, temos que

$$\log n < n.$$

Daqui segue que

$$\log n \text{ é } O(n).$$

(Novamente aceitamos $C = k = 1$ como parâmetros.)

Se temos logaritmos de base b, em que b é diferente de 2, temos ainda que $\log_b n$ é $O(n)$, pois

$$\log_b n = \frac{\log n}{\log b} < \frac{n}{\log b}$$

sempre que n for um número inteiro positivo. Tomamos $C = 1/\log b$ e $k = 1$ como parâmetros. (Usamos o Teorema 3 do Apêndice 2 para verificar que $\log_b n = \log n / \log b$.) ◀

Como foi mencionado antes, a notação O grande é usada para estimar o número de operações necessárias para resolver um problema usando-se um determinado procedimento ou algoritmo. As funções usadas nessas estimativas geralmente incluem as seguintes:

$$1, \log n, n, n \log n, n^2, 2^n, n!$$

Usando cálculo, é possível mostrar que cada função da lista é menor que a função seguinte, no sentido de que a razão de uma função e a função seguinte tende a zero à medida que n cresce ilimitadamente. A Figura 3 apresenta o gráfico dessas funções usando uma escala para os valores das funções, que dobram para cada marca sucessiva do gráfico. Ou seja, a escala vertical no gráfico é logarítmica.

O Crescimento de Combinações de Funções

Muitos algoritmos são feitos com dois ou mais procedimentos separados. O número de passos usados por um computador para resolver um problema com uma entrada de determinado tamanho usando um algoritmo é a soma do número de passos usados por esses subprocedimentos. Para dar uma estimativa O grande para o número de passos necessários, é preciso encontrar as estimativas O grande para o número de passos usados pelos subprocedimentos e então combinar essas estimativas.

As estimativas O grande da combinação de funções podem ser fornecidas se for tomado cuidado quando diferentes estimativas O grande forem combinadas. Em particular, geralmente é necessário estimar o crescimento da soma e do produto de duas funções. O que pode ser dito se as estimativas O grande para cada uma das duas funções forem conhecidas? Para ver qual tipo de

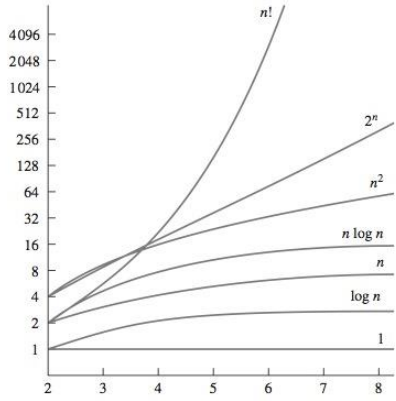


FIGURA 3 Uma Demonstração do Crescimento das Funções geralmente Usadas em Estimativas *O* grande.

estimativa é usada para a soma e para o produto de duas funções, suponha que $f_1(x)$ seja $O(g_1(x))$ e $f_2(x)$ seja $O(g_2(x))$.

A partir da definição de notação O grande, há constantes C_1, C_2, k_1 e k_2, tal que

$$|f_1(x)| \leq C_1|g_1(x)|$$

quando $x > k_1$, e

$$|f_2(x)| \leq C_2|g_2(x)|$$

quando $x > k_2$. Para estimar a soma de $f_1(x)$ e $f_2(x)$, note que

$$\begin{aligned} |(f_1 + f_2)(x)| &= |f_1(x) + f_2(x)| \\ &\leq |f_1(x)| + |f_2(x)| \qquad \text{usando a inequação triangular } |a+b| \leq |a| + |b|. \end{aligned}$$

Quando x é maior que k_1 e k_2, temos, a partir das inequações para $|f_1(x)|$ e $|f_2(x)|$, que

$$\begin{aligned} |f_1(x)| + |f_2(x)| &\leq C_1|g_1(x)| + C_2|g_2(x)| \\ &\leq C_1|g(x)| + C_2|g(x)| \\ &= (C_1 + C_2)|g(x)| \\ &= C|g(x)|, \end{aligned}$$

em que $C = C_1 + C_2$ e $g(x) = \max(|g_1(x)|, |g_2(x)|)$. [Aqui, $\max(a, b)$ indica o máximo, ou maior, entre a e b.]

Esta inequação mostra que $|(f_1 + f_2)(x)| \leq C\,|g(x)|$ sempre que $x > k$, em que $k = \max(k_1, k_2)$. Constatamos este resultado útil como Teorema 2.

TEOREMA 2 Suponha que $f_1(x)$ seja $O(g_1(x))$ e $f_2(x)$ seja $O(g_2(x))$. Então, $(f_1 + f_2)(x)$ é $O(\max(|g_1(x)|, |g_2(x)|))$.

Geralmente temos estimativas O grande para f_1 e f_2 em termos da mesma função g. Neste caso, o Teorema 2 pode ser usado para mostrar que $(f_1 + f_2)(x)$ é também $O(g(x))$, porque $\max(g(x), g(x)) = g(x)$. Este resultado é confirmado pelo Corolário 1.

COROLÁRIO 1 Suponha que $f_1(x)$ e $f_2(x)$ sejam $O(g(x))$. Então, $(f_1 + f_2)(x)$ é $O(g(x))$.

De forma semelhante, as estimativas O grande podem ser derivadas para o produto das funções f_1 e f_2. Quando x é maior que $\max(k_1, k_2)$, temos que

$$\begin{aligned} |(f_1 f_2)(x)| &= |f_1(x)||f_2(x)| \\ &\leq C_1|g_1(x)|C_2|g_2(x)| \\ &\leq C_1C_2|(g_1g_2)(x)| \\ &\leq C\,|(g_1g_2)(x)|, \end{aligned}$$

em que $C = C_1C_2$. A partir desta inequação, temos que $f_1(x)f_2(x)$ é $O(g_1g_2)$, pois há constantes C e k, com $C = C_1C_2$ e $k = \max(k_1, k_2)$, tal que $|(f_1 f_2)(x)| \leq C\,|g_1(x)g_2(x)|$ sempre que $x > k$. Este resultado é confirmado pelo Teorema 3.

TEOREMA 3 Suponha que $f_1(x)$ seja $O(g_1(x))$ e $f_2(x)$ seja $O(g_2(x))$. Então, $(f_1 f_2)(x)$ é $O(g_1(x)g_2(x))$.

O objetivo de usar a notação O grande para estimar funções é escolher uma função $g(x)$ que cresce relativamente devagar para que $f(x)$ seja $O(g(x))$. Os exemplos 8 e 9 mostram como usar os Teoremas 2 e 3 para fazer isso. O tipo de análise mostrado nesses exemplos é geralmente usado na análise do tempo gasto para resolver problemas usando programas de computador.

EXEMPLO 8 Dê uma estimativa O grande para $f(n) = 3n \log(n!) + (n^2 + 3) \log n$, em que n é um número inteiro positivo.

Solução: Primeiro, o produto $3n \log(n!)$ será estimado. A partir do Exemplo 6, sabemos que $\log(n!)$ é $O(n \log n)$. Usando essa estimativa e o fato de que $3n$ é $O(n)$, o Teorema 3 estima que $3n \log(n!)$ seja $O(n^2 \log n)$.

Depois, o produto $(n^2 + 3) \log n$ será estimado. Como $(n^2 + 3) < 2n^2$ quando $n > 2$, temos que $n^2 + 3$ é $O(n^2)$. Então, a partir do Teorema 3, temos que $(n^2 + 3) \log n$ é $O(n^2 \log n)$. Usando o Teorema 2 para combinar as duas estimativas O grande para os produtos, temos que $f(n) = 3n \log(n!) + (n^2 + 3) \log n$ é $O(n^2 \log n)$. ◀

EXEMPLO 9 Dê uma estimativa O grande para $f(x) = (x + 1) \log(x^2 + 1) + 3x^2$.

Solução: Primeiro, encontraremos uma estimativa O grande para $(x + 1) \log(x^2 + 1)$. Note que $(x + 1)$ é $O(x)$. Além disso, $x^2 + 1 \leq 2x^2$ quando $x > 1$. Assim,

$$\log(x^2 + 1) \leq \log(2x^2) = \log 2 + \log x^2 = \log 2 + 2 \log x \leq 3 \log x,$$

se $x > 2$. Isto mostra que $\log(x^2 + 1)$ é $O(\log x)$.

A partir do Teorema 3, temos que $(x + 1)\log(x^2 + 1)$ é $O(x \log x)$. Como $3x^2$ é $O(x^2)$, o Teorema 2 nos diz que $f(x)$ é $O(\max(x \log x, x^2))$. Como $x \log x \leq x^2$, para $x > 1$, temos que $f(x)$ é $O(x^2)$. ◀

Notações Ômega grande e Teta grande

A notação O grande é muito utilizada para descrever o crescimento de funções, mas tem suas limitações. Em particular, quando $f(x)$ é $O(g(x))$, temos uma limitante superior, em termos de $g(x)$, para o tamanho de $f(x)$ para valores maiores de x. Entretanto, a notação O grande não fornece uma limitante inferior para o tamanho de $f(x)$ para um valor maior de x. Para isso, usamos a **notação Ômega grande (Ω grande).** Quando queremos dar uma limitante superior e uma inferior ao tamanho de uma função $f(x)$, relativa à função referencial $g(x)$, usamos a **notação Teta grande (Θ grande).** Ambas as notações, Ômega e Teta, foram introduzidas por Donald Knuth na década de 1970. Sua motivação para introduzir essas notações foi a não-utilização da notação O grande quando uma limitante superior e uma inferior em relação ao tamanho de uma função eram necessárias.

Definimos agora a notação Ômega grande e mostraremos seu uso. Depois de fazer isto, faremos o mesmo com a notação Teta grande.

Há uma forte ligação entre as notações O grande e Ômega grande. Particularmente, $f(x)$ é $\Omega(g(x))$ se e somente se $g(x)$ for $O(f(x))$. Deixaremos a verificação deste fato como exercício para o leitor.

DEFINIÇÃO 2 Considere f e g como funções do conjunto de números inteiros ou do conjunto de números reais para o conjunto de números reais. Dizemos que $f(x)$ é $\Omega(g(x))$ se houver constantes positivas C e k, tal que

$$|f(x)| \geq C|g(x)|$$

sempre que $x > k$. [Isto é lido como "$f(x)$ é Ômega grande de $g(x)$."]

EXEMPLO 10 A função $f(x) = 8x^3 + 5x^2 + 7$ é $\Omega(g(x))$, em que $g(x)$ é a função $g(x) = x^3$. É fácil verificar isto, pois $f(x) = 8x^3 + 5x^2 + 7 \geq 8x^3$ para todo número real positivo x. É o mesmo que dizer que $g(x) = x^3$ é $O(8x^3 + 5x^2 + 7)$, que pode ser estabelecido diretamente virando a inequação. ◀

É importante saber a ordem do crescimento de uma função em termos de alguma função referencial relativamente simples, tal como x^n quando n for um número inteiro positivo ou c^x, em que $c > 1$. Saber a ordem do crescimento requer que tenhamos uma limitante superior e uma inferior para o tamanho da função. Ou seja, dada uma função $f(x)$, queremos a função referente $g(x)$, tal que $f(x)$ é $O(g(x))$ e $f(x)$ é $\Omega(g(x))$. A notação Teta grande, definida a seguir, é usada para expressar essas relações, fornecendo uma limitante superior e uma inferior para o tamanho de uma função.

DEFINIÇÃO 3 Considere f e g como funções do conjunto de números inteiros ou números reais para o conjunto de números reais. Dizemos que $f(x)$ é $\Theta(g(x))$, se $f(x)$ for $O(g(x))$ e $f(x)$ for $\Omega(g(x))$. Quando $f(x)$ é $\Theta(g(x))$, dizemos que "f é Teta grande de $g(x)$" e também dizemos que $f(x)$ é da *ordem* de $g(x)$.

Quando $f(x)$ é $\Theta(g(x))$, é também o caso de $g(x)$ ser $\Theta(f(x))$. Note também que $f(x)$ é $\Theta(g(x))$ se e somente se $f(x)$ for $O(g(x))$ e $g(x)$ for $O(f(x))$ (veja o Exercício 25).

Geralmente, quando a notação Teta grande é usada, a função $g(x)$ em $\Theta(g(x))$ é uma função referencial relativamente simples, assim como x^n, c^x, $\log x$ e assim por diante, enquanto $f(x)$ pode ser relativamente complicada.

EXEMPLO 11 Mostramos (no Exemplo 5) que a soma dos primeiros n números inteiros positivos é $O(n^2)$. Esta soma é de ordem n^2?

Exemplos Extras

Solução: Considere $f(n) = 1 + 2 + 3 + \cdots + n$. Como já sabemos que $f(n)$ é $O(n^2)$, para mostrar que $f(n)$ é de ordem n^2, precisamos encontrar uma constante positiva C, tal que $f(n) > Cn^2$ para n números inteiros suficientemente grandes. Para obter uma limitante inferior para esta soma, podemos ignorar a primeira metade dos termos. Somando apenas os termos maiores que $\lceil n/2 \rceil$, encontramos que

$$\begin{aligned} 1 + 2 + \cdots + n &\geq \lceil n/2 \rceil + (\lceil n/2 \rceil + 1) + \cdots + n \\ &\geq \lceil n/2 \rceil + \lceil n/2 \rceil + \cdots + \lceil n/2 \rceil \\ &= (n - \lceil n/2 \rceil + 1)\lceil n/2 \rceil \\ &\geq (n/2)(n/2) \\ &= n^2/4. \end{aligned}$$

Isto mostra que $f(n)$ é $\Omega(n^2)$. Concluímos que $f(n)$ é de ordem n^2, ou, em símbolos, $f(n)$ é $\Theta(n^2)$. ◀

Podemos mostrar que $f(x)$ é $\Theta(g(x))$ se conseguirmos encontrar números reais positivos C_1 e C_2 e um número real positivo k tal que

$$C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$$

sempre que $x > k$. Isto mostra que $f(x)$ é $O(g(x))$ e $f(x)$ é $\Omega(g(x))$.

EXEMPLO 12 Mostre que $3x^2 + 8x \log x$ é $\Theta(x^2)$.

Exemplos Extras

Solução: Como $0 \leq 8x \log x \leq 8x^2$, temos que $3x^2 + 8x \log x \leq 11x^2$ para $x > 1$. Conseqüentemente, $3x^2 + 8x \log x$ é $O(x^2)$. Claramente, x^2 é $O(3x^2 + 8x \log x)$. Conseqüentemente, $3x^2 + 8x \log x$ é $\Theta(x^2)$. ◀

Um fato útil é que o termo principal de um polinômio determina sua ordem. Por exemplo, se $f(x) = 3x^5 + x^4 + 17x^3 + 2$, então $f(x)$ é de ordem x^5. Isto é confirmado pelo Teorema 4, cuja demonstração foi deixada como um exercício no final desta seção.

TEOREMA 4 Considere $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, em que $a_0, a_1, \ldots, a_n$ são números reais com $a_n \neq 0$. Então, $f(x)$ é de ordem x^n.

EXEMPLO 13 Os polinômios $3x^8 + 10x^7 + 221x^2 + 1444$, $x^{19} - 18x^4 - 10{,}112$ e $-x^{99} + 40{,}001x^{98} + 100{,}003x$ são de ordem x^8, x^{19} e x^{99}, respectivamente. ◀

Infelizmente, como Knuth observou, a notação O grande é normalmente utilizada por escritores sem cuidado como se tivesse o mesmo significado da notação Teta grande. Mantenha isto em mente quando você vir a notação O grande sendo usada. A recente tendência tem sido na direção de usar a notação Teta grande sempre que forem necessárias uma limitante superior e uma inferior em relação ao tamanho de uma função.

Exercícios

Nos exercícios de 1 a 14, para estabelecer uma relação O grande, encontre os parâmetros C e k, tal que $|f(x)| \leq C\,|g(x)|$ sempre que $x > k$.

1. Determine se cada uma destas funções é $O(x)$.
   a) $f(x) = 10$ **b)** $f(x) = 3x + 7$
   c) $f(x) = x^2 + x + 1$ **d)** $f(x) = 5 \log x$
   e) $f(x) = \lfloor x \rfloor$ **f)** $f(x) = \lceil x/2 \rceil$
2. Determine se cada uma destas funções é $O(x^2)$.
   a) $f(x) = 17x + 11$ **b)** $f(x) = x^2 + 1000$
   c) $f(x) = x \log x$ **d)** $f(x) = x^4/2$
   e) $f(x) = 2^x$ **f)** $f(x) = \lfloor x \rfloor \cdot \lceil x \rceil$
3. Use a definição de "$f(x)$ é $O(g(x))$" para mostrar que $x^4 + 9x^3 + 4x + 7$ é $O(x^4)$.
4. Use a definição de "$f(x)$ é $O(g(x))$" para mostrar que $2^x + 17$ is $O(3^x)$.
5. Mostre que $(x^2 + 1)/(x + 1)$ é $O(x)$.
6. Mostre que $(x^3 + 2x)/(2x + 1)$ é $O(x^2)$.
7. Encontre o menor número inteiro n para que $f(x)$ seja $O(x^n)$ para cada uma destas funções.
   a) $f(x) = 2x^3 + x^2 \log x$
   b) $f(x) = 3x^3 + (\log x)^4$
   c) $f(x) = (x^4 + x^2 + 1)/(x^3 + 1)$
   d) $f(x) = (x^4 + 5 \log x)/(x^4 + 1)$
8. Encontre o menor número inteiro n para que $f(x)$ seja $O(x^n)$ para cada uma destas funções.
   a) $f(x) = 2x^2 + x^3 \log x$
   b) $f(x) = 3x^5 + (\log x)^4$
   c) $f(x) = (x^4 + x^2 + 1)/(x^4 + 1)$
   d) $f(x) = (x^3 + 5 \log x)/(x^4 + 1)$
9. Mostre que $x^2 + 4x + 17$ é $O(x^3)$, mas que x^3 não é $O(x^2 + 4x + 17)$.
10. Mostre que x^3 é $O(x^4)$, mas que x^4 não é $O(x^3)$.
11. Mostre que $3x^4 + 1$ é $O(x^4/2)$ e $x^4/2$ é $O(3x^4 + 1)$.
12. Mostre que $x \log x$ é $O(x^2)$, mas que x^2 não é $O(x \log x)$.
13. Mostre que 2^n é $O(3^n)$, mas que 3^n não é $O(2^n)$.
14. É verdade que x^3 é $O(g(x))$, se g é a função dada? [Por exemplo, se $g(x) = x + 1$, esta questão pergunta se x^3 é $O(x + 1)$.]
   a) $g(x) = x^2$ **b)** $g(x) = x^3$
   c) $g(x) = x^2 + x^3$ **d)** $g(x) = x^2 + x^4$
   e) $g(x) = 3^x$ **f)** $g(x) = x^3/2$
15. Explique o que significa uma função ser $O(1)$.
16. Mostre que se $f(x)$ é $O(x)$, então $f(x)$ é $O(x^2)$.
17. Suponha que $f(x)$, $g(x)$ e $h(x)$ são funções, tal que $f(x)$ é $O(g(x))$ e $g(x)$ é $O(h(x))$. Mostre que $f(x)$ é $O(h(x))$.
18. Considere k como um número inteiro positivo. Mostre que $1^k + 2^k + \cdots + n^k$ é $O(n^{k+1})$.
19. Dê a melhor estimativa O grande possível para cada uma destas funções.
   a) $(n^2 + 8)(n + 1)$ **b)** $(n \log n + n^2)(n^3 + 2)$
   c) $(n! + 2^n)(n^3 + \log(n^2 + 1))$
20. Dê uma estimativa O grande para cada uma destas funções. Como a função g em sua estimativa $f(x)$ é $O(g)$, use uma função simples g de menor ordem.
   a) $(n^3 + n^2 \log n)(\log n + 1) + (17 \log n + 19)(n^3 + 2)$
   b) $(2^n + n^2)(n^3 + 3^n)$
   c) $(n^n + n2^n + 5^n)(n! + 5^n)$
21. Dê uma estimativa O grande para cada uma destas funções. Como a função g em sua estimativa $f(x)$ é $O(g(x))$, use uma função simples g de menor ordem.
   a) $n \log(n^2 + 1) + n^2 \log n$
   b) $(n \log n + 1)^2 + (\log n + 1)(n^2 + 1)$
   c) $n^{2n} + n^{n^2}$
22. Para cada função do Exercício 1, determine se ela é $\Omega(x)$ e se é $\Theta(x)$.
23. Para cada função do Exercício 2, determine se ela é $\Omega(x^2)$ e se é $\Theta(x^2)$.
24. **a)** Mostre que $3x + 7$ é $\Theta(x)$.
   b) Mostre que $2x^2 + x - 7$ é $\Theta(x^2)$.
   c) Mostre que $\lfloor x + 1/2 \rfloor$ é $\Theta(x)$.
   d) Mostre que $\log(x^2 + 1)$ é $\Theta(\log_2 x)$.
   e) Mostre que $\log_{10} x$ é $\Theta(\log_2 x)$.
25. Mostre que $f(x)$ é $\Theta(g(x))$ se e somente se $f(x)$ for $O(g(x))$ e $g(x)$ for $O(f(x))$.
26. Mostre que se $f(x)$ e $g(x)$ forem funções do conjunto de números reais para o conjunto de números reais, então $f(x)$ é $O(g(x))$ se e somente se $g(x)$ for $\Omega(f(x))$.
27. Mostre que se $f(x)$ e $g(x)$ forem funções do conjunto de números reais para o conjunto de números reais, então $f(x)$ é $\Theta(g(x))$ se e somente se houver constantes positivas k, C_1 e C_2 tal que $C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$ sempre que $x > k$.
28. **a)** Mostre que $3x^2 + x + 1$ é $\Theta(3x^2)$ encontrando diretamente as constantes k, C_1 e C_2 do Exercício 27.
   b) Expresse a relação do item (a) usando um esquema que mostre as funções $3x^2 + x + 1$, $C_1 \cdot 3x^2$ e $C_2 \cdot 3x^2$ e a constante k no eixo x, em que C_1, C_2 e k são as constantes que você encontrou no item (a), para mostrar que $3x^2 + x + 1$ é $\Theta(3x^2)$.
29. Expresse a relação em que $f(x)$ é $\Theta(g(x))$ usando um esquema. Mostre os gráficos das funções $f(x)$, $C_1|g(x)|$ e $C_2|g(x)|$, assim como a constante k no eixo x.
30. Explique o que significa uma função ser $\Omega(1)$.
31. Explique o que significa uma função ser $\Theta(1)$.
32. Dê uma estimativa O grande para o produto dos primeiros n números inteiros positivos ímpares.
33. Mostre que se f e g são funções com valor real, tal que $f(x)$ é $O(g(x))$, então, para cada número inteiro positivo k, $f^k(x)$ é $O(g^k(x))$. [Note que $f^k(x) = f(x)^k$.]
34. Mostre que para todos os números reais a e b com $a > 1$ e $b > 1$, se $f(x)$ é $O(\log_b x)$, então $f(x)$ é $O(\log_a x)$.

35. Suponha que $f(x)$ seja $O(g(x))$, em que f e g são funções crescentes e ilimitadas. Mostre que $\log |f(x)|$ é $O(\log |g(x)|)$.

36. Suponha que $f(x)$ seja $O(g(x))$. É verdade que $2^{f(x)}$ é $O(2^{g(x)})$?

37. Considere $f_1(x)$ e $f_2(x)$ como funções do conjunto de números reais para o conjunto de números reais positivos. Mostre que se $f_1(x)$ e $f_2(x)$ são $\Theta(g(x))$, em que $g(x)$ é uma função do conjunto dos números reais para o conjunto dos números reais positivos, então $f_1(x) + f_2(x)$ é $\Theta(g(x))$. Isto ainda é verdadeiro se $f_1(x)$ e $f_2(x)$ puderem ter valores negativos?

38. Suponha que $f(x)$, $g(x)$ e $h(x)$ são funções, tal que $f(x)$ é $\Theta(g(x))$ e $g(x)$ é $\Theta(h(x))$. Mostre que $f(x)$ é $\Theta(h(x))$.

39. Se $f_1(x)$ e $f_2(x)$ são funções do conjunto de números inteiros positivos para o conjunto de números reais positivos e $f_1(x)$ e $f_2(x)$ são $\Theta(g(x))$, $(f_1 - f_2)(x)$ também é $\Theta(g(x))$? Demonstre que é ou dê um contra-exemplo.

40. Mostre que se $f_1(x)$ e $f_2(x)$ são funções do conjunto de números inteiros positivos para o conjunto de números reais e $f_1(x)$ é $\Theta(g_1(x))$ e $f_2(x)$ é $\Theta(g_2(x))$, então $(f_1f_2)(x)$ é $\Theta(g_1g_2(x))$.

41. Encontre funções f e g do conjunto de números inteiros positivos para o conjunto de números reais, tal que $f(n)$ não é $O(g(n))$ e $g(n)$ não é $O(f(n))$ simultaneamente.

42. Expresse a relação em que $f(x)$ é $\Omega(g(x))$ usando um esquema. Mostre os gráficos das funções $f(x)$ e $Cg(x)$, assim como a constante k no eixo dos reais.

43. Mostre que se $f_1(x)$ é $\Theta(g_1(x))$, $f_2(x)$ é $\Theta(g_2(x))$ e $f_2(x) \neq 0$ e $g_2(x) \neq 0$ para todos os números reais $x > 0$, então $(f_1/f_2)(x)$ é $\Theta((g_1/g_2)(x))$.

44. Mostre que se $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0$, em que $a_0, a_1, \ldots, a_{n-1}$ e a_n são números reais e $a_n \neq 0$, então $f(x)$ é $\Theta(x^n)$.

As notações O grande, Teta grande e Ômega grande podem ser estendidas a funções com mais de uma variável. Por exemplo, a proposição de que $f(x, y)$ é $O(g(x, y))$ significa que existem as constantes C, k_1 e k_2, tal que $|f(x, y)| \leq C\,|g(x, y)|$ sempre que $x > k_1$ e $y > k_2$.

45. Defina a proposição de que $f(x, y)$ é $\Theta(g(x, y))$.

46. Defina a proposição de que $f(x, y)$ é $\Omega(g(x, y))$.

47. Mostre que $(x^2 + xy + x \log y)^3$ é $O(x^6\, y^3)$.

48. Mostre que $x^5\, y^3 + x^4\, y^4 + x^3\, y^5$ é $\Omega(x^3\, y^3)$.

49. Mostre que $\lfloor xy \rfloor$ é $O(xy)$.

50. Mostre que $\lceil xy \rceil$ é $\Omega(xy)$.

Os problemas a seguir lidam com outro tipo de notação assintótica, chamada de ***o* pequeno**. Como ela é baseada no conceito de limites, um conhecimento em cálculo é necessário para estes problemas. Dizemos que $f(x)$ é $o(g(x))$ [leia-se $f(x)$ é o pequeno de $g(x)$], quando

$$\lim_{x \to \infty} \frac{f(x)}{g(x)} = 0.$$

51. (*Requer cálculo*) Mostre que

a) x^2 é $o(x^3)$. **b)** $x \log x$ é $o(x^2)$.

c) x^2 é $o(2^x)$. **d)** $x^2 + x + 1$ não é $o(x^2)$.

52. (*Requer cálculo*)

a) Mostre que se $f(x)$ e $g(x)$ são funções, tal que $f(x)$ é $o(g(x))$ e c é uma constante, então $cf(x)$ é $o(g(x))$, em que $(cf)(x) = cf(x)$.

b) Mostre que se $f_1(x), f_2(x)$ e $g(x)$ são funções, tal que $f_1(x)$ é $o(g(x))$ e $f_2(x)$ é $o(g(x))$, então $(f_1 + f_2)(x)$ é $o(g(x))$, em que $(f_1 + f_2)(x) = f_1(x) + f_2(x)$.

53. (*Requer cálculo*) Represente esquematicamente que $x \log x$ é $o(x^2)$ pelo gráfico de $x \log x$, x^2 e $x \log x/x^2$. Explique como este esquema mostra que $x \log x$ é $o(x^2)$.

54. (*Requer cálculo*) Expresse a relação em que $f(x)$ é $o(g(x))$ usando um esquema. Mostre os gráficos de $f(x)$, $g(x)$ e $f(x)/g(x)$.

***55.** (*Requer cálculo*) Suponha que $f(x)$ seja $o(g(x))$. É verdade que $2^{f(x)}$ é $o(2^{g(x)})$?

***56.** (*Requer cálculo*) Suponha que $f(x)$ seja $o(g(x))$. É verdade que $\log |f(x)|$ é $o(\log |g(x)|)$?

57. (*Requer cálculo*) As duas partes deste exercício descrevem a relação entre as notações o pequeno e O grande.

a) Mostre que se $f(x)$ e $g(x)$ são funções, tal que $f(x)$ é $o(g(x))$, então $f(x)$ é $O(g(x))$.

b) Mostre que se $f(x)$ e $g(x)$ são funções, tal que $f(x)$ é $O(g(x))$, então não é necessariamente verdade que $f(x)$ seja $o(g(x))$.

58. (*Requer cálculo*) Mostre que se $f(x)$ é um polinômio de grau n e $g(x)$ é um polinômio de grau m, em que $m > n$, então $f(x)$ é $o(g(x))$.

59. (*Requer cálculo*) Mostre que se $f_1(x)$ é $O(g(x))$ e $f_2(x)$ é $o(g(x))$, então $f_1(x) + f_2(x)$ é $O(g(x))$.

60. (*Requer cálculo*) Considere H_n como o n-ésimo **número harmônico**

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}.$$

Mostre que H_n é $O(\log n)$. [*Dica:* Primeiro estabeleça a inequação

$$\sum_{j=2}^{n} \frac{1}{j} < \int_1^n \frac{1}{x}\,dx$$

mostrando que a soma das áreas dos retângulos de altura $1/j$ com base de $j - 1$ a j, para $j = 2, 3, \ldots, n$, é menor que a área da curva $y = 1/x$ de 2 a n.]

***61.** Mostre que $n \log n$ é $O(\log n!)$.

☞**62.** Determine se $\log n!$ é $\Theta(n \log n)$. Justifique sua resposta.

***63.** Mostre que $\log n!$ é maior que $(n \log n)/4$ para $n > 4$. [*Dica:* Comece com a inequação $n! > n(n-1)\,(n-2) \cdots \lceil n/2 \rceil$.]

Considere $f(x)$ e $g(x)$ como funções do conjunto de números reais para o conjunto de números reais. Dizemos que as funções f e g são **assintóticas** e escrevemos $f(x) \sim g(x)$ se $\lim_{x \to \infty} f(x)/g(x) = 1$.

64. (*Requer cálculo*) Para cada um destes pares de funções, determine se f e g são assintóticas.

a) $f(x) = x^2 + 3x + 7$, $g(x) = x^2 + 10$

b) $f(x) = x^2 \log x$, $g(x) = x^3$
c) $f(x) = x^4 + \log(3x^8 + 7)$, $g(x) = (x^2 + 17x + 3)^2$
d) $f(x) = (x^3 + x^2 + x + 1)^4$, $g(x) = (x^4 + x^3 + x^2 + x + 1)^3$.

65. (*Requer cálculo*) Para cada um destes pares de funções, determine se f e g são assintóticas.
a) $f(x) = \log(x^2 + 1)$, $g(x) = \log x$
b) $f(x) = 2^{x+3}$, $g(x) = 2^{x+7}$
c) $f(x) = 2^{2x}$, $g(x) = 2^{x^2}$
d) $f(x) = 2^{x^2+x+1}$, $g(x) = 2^{x^2+2x}$

3.3 Complexidade dos Algoritmos

Introdução

O que é necessário para um algoritmo fornecer uma solução satisfatória para um problema? Primeiro, ele deve sempre produzir uma resposta correta. Discutiremos no Capítulo 4 como isso pode ser demonstrado. Segundo, ele deve ser eficiente. A eficiência dos algoritmos será discutida nesta seção.

Como a eficiência de um algoritmo pode ser analisada? Uma medida de eficiência é o tempo de computação necessário para resolver um problema usando o algoritmo, quando os valores de entrada são de determinado tamanho. Uma segunda medida é a quantidade de memória computacional necessária para implementar o algoritmo quando os valores de entrada forem de tamanho determinado.

Questões como essas envolvem a **complexidade computacional** do algoritmo. Uma análise do tempo necessário para resolver um problema de um determinado tamanho envolve a **complexidade temporal** do algoritmo. Uma análise da memória computacional necessária envolve a **complexidade espacial** do algoritmo. As considerações sobre complexidade temporal e espacial de um algoritmo são essenciais quando eles são implementados. É óbvia a importância de saber se um algoritmo irá produzir uma resposta em um microssegundo, em um minuto ou em bilhões de anos. Da mesma forma, a memória necessária deve estar disponível para resolver um problema, portanto é preciso que a complexidade espacial seja levada em conta.

As considerações sobre complexidade espacial estão ligadas a determinadas estruturas de dados usadas para implementar o algoritmo. Como essas estruturas não são abordadas detalhadamente neste livro, a complexidade espacial não será considerada. Assim, vamos restringir nossa atenção à complexidade temporal.

Complexidade Temporal

A complexidade temporal de um algoritmo pode ser expressa em termos de números de operações usadas por um algoritmo quando a entrada tem um determinado tamanho. As operações usadas para medir a complexidade temporal podem ser a comparação, a adição, a multiplicação e a divisão de números inteiros ou qualquer outra operação básica.

A complexidade temporal é descrita em termos do número de operações necessárias em vez do tempo atual do computador, por causa da diferença de tempo necessária para computadores diferentes realizarem as operações básicas. Além disso, é um pouco difícil quebrar todas as operações binárias que um computador usa. Além do mais, os computadores mais rápidos podem realizar as operações binárias básicas (por exemplo, adição, multiplicação, comparação ou troca de dois bits) em 10^{-9} segundos (1 nanossegundo), mas os computadores domésticos podem precisar de 10^{-6} segundo (1 microssegundo), que é um tempo 1 000 vezes mais longo, para fazer as mesmas operações.

Mostraremos agora como analisar a complexidade temporal de um algoritmo considerando o Algoritmo 1 da Seção 3.1, que encontra o maior elemento em um conjunto finito de números inteiros.

EXEMPLO 1 Descreva a complexidade temporal do Algoritmo 1 da Seção 3.1 para encontrar o maior elemento em um conjunto.

Exemplos Extras

Solução: O número de comparações será usado como medida da complexidade temporal do algoritmo, porque comparações são as operações básicas usadas.

Para encontrar o maior elemento de um conjunto de n elementos, listados em ordem arbitrária, o máximo temporário do conjunto é igual ao primeiro termo da lista. Então, depois que uma comparação foi feita para verificar se o final da lista não foi alcançado, o máximo temporário e um segundo termo são comparados, atualizando o máximo temporário para o valor do segundo termo, se este for maior. Continuamos este procedimento usando duas comparações adicionais para cada termo da lista — uma para determinar se o final da lista não foi alcançado e outra para determinar a atualização do máximo temporário. Como duas comparações são usadas para cada um dos elementos do segundo termo até o n-ésimo e mais uma comparação é usada para finalizar o laço quando $i = n + 1$, exatamente $2(n - 1) + 1 = 2n - 1$ comparações são usadas se este algoritmo for aplicado. Assim, o algoritmo para encontrar o maior elemento em um conjunto com n elementos tem complexidade temporal $\Theta(n)$, medida em termos do número de comparações usadas. ◀

A seguir, analisaremos a complexidade temporal dos algoritmos de busca.

EXEMPLO 2 Descreva a complexidade temporal do algoritmo de busca linear.

Solução: O número de comparações usadas pelo algoritmo será tomado como medida da complexidade temporal. A cada passo do laço no algoritmo, duas comparações são realizadas — uma para ver se o final da lista foi alcançado e outra para comparar o elemento x com um termo da lista. Por fim, uma comparação a mais é feita por fora do laço. Conseqüentemente, se $x = a_i$, $2i + 1$ comparações são usadas. As maiores comparações, $2n + 2$, são necessárias quando o elemento não está na lista. Neste caso, $2n$ comparações são usadas para determinar que x não é a_i, para $i = 1, 2, \ldots, n$, uma comparação adicional é usada para finalizar o laço e uma comparação é feita por fora do laço. Então, quando x não estiver na lista, um total de $2n + 2$ comparações são usadas. Assim, uma busca linear requer no máximo $O(n)$ comparações. ◀

COMPLEXIDADE DE PIOR CASO O tipo de análise de complexidade do Exemplo 2 é uma análise do **pior caso**. Por análise do pior caso de um algoritmo, entendemos, o maior número de operações necessárias para resolver o problema dado usando esse algoritmo com uma entrada de determinado tamanho. A análise do pior caso nos diz quantas operações um algoritmo necessita para garantir que irá produzir uma solução.

EXEMPLO 3 Descreva a complexidade temporal do algoritmo de busca binária em termos do número de comparações usadas (e ignore o tempo necessário para computar $m = [(i + j)/2]$ em cada iteração do laço no algoritmo).

Solução: Simplificando, assuma que há $n = 2^k$ elementos na lista $a_1, a_2, \ldots, a_n$, em que k é um número inteiro não negativo. Note que $k = \log n$. (Se n, o número de elementos na lista, não for uma potência de 2, a lista pode ser considerada como parte de uma lista maior com 2^{k+1} elementos, em que $2^k < n < 2^{k+1}$. Assim, 2^{k+1} é a menor potência de 2 maior que n.)

Em cada etapa do algoritmo, i e j, as localizações do primeiro e do último termo da lista restrita em cada etapa são comparadas para ver se a lista restrita tem mais de um termo. Se $i < j$, uma comparação é feita para determinar se x é maior que o termo do meio dessa lista.

A primeira etapa da busca é restrita à lista com 2^{k-1} termos. Até aqui, duas comparações foram feitas. Este procedimento continua, usando duas comparações em cada etapa para restringir a busca a uma lista com metade dos termos. Em outras palavras, duas comparações são usadas na primeira etapa do algoritmo, quando a lista tem 2^k elementos, mais duas quando a busca foi reduzida à lista com 2^{k-1} elementos, mais duas quando a busca foi reduzida à lista com 2^{k-2} elementos, e assim por diante, até que duas comparações sejam usadas quando a busca for reduzida à lista com $2^1 = 2$ elementos. Por fim, quando um termo estiver à esquerda na lista, uma comparação nos diz que não há termos adicionais restantes, e mais uma comparação é usada para determinar se este termo é x.

Assim, pelo menos $2k + 2 = 2 \log n + 2$ comparações são necessárias para realizar uma busca binária quando a lista a ser pesquisada tiver 2^k elementos. (Se n não for uma potência de 2, a lista original é expandida para uma lista com 2^{k+1} termos, em que $k = \lfloor \log n \rfloor$, e a pesquisa requer, no máximo, $2\lceil \log n \rceil + 2$ comparações.) Conseqüentemente, uma busca binária necessita de, no máximo, $\Theta(\log n)$ comparações. A partir desta análise, temos que o algoritmo de busca binária é mais eficiente, no pior caso, que uma busca linear. ◀

COMPLEXIDADE DE CASO MÉDIO Outro tipo importante de análise de complexidade, além da análise do pior caso, é chamada de análise do **caso médio**. O número médio de operações usadas para resolver os problemas sobre todas as entradas de determinados tamanhos é encontrado neste tipo de análise. A complexidade temporal de caso médio é geralmente muito mais complicada que a análise do pior caso. Entretanto, a análise do caso médio para o algoritmo de busca linear pode ser feita sem muitas dificuldades, como mostra o Exemplo 4.

EXEMPLO 4 Descreva a performance de caso médio do algoritmo de busca linear, supondo que o elemento x está na lista.

Solução: Há n tipos de entradas possíveis quando se sabe que x está na lista. Se x for o primeiro termo da lista, são necessárias três comparações: uma para determinar se o final da lista foi alcançado, uma para comparar x com o primeiro termo e uma por fora do laço. Se x for o segundo termo da lista, são necessárias mais duas comparações, portanto são usadas cinco comparações no total. No geral, se x for o i-ésimo termo da lista, duas comparações serão usadas para cada um dos i passos do laço e uma por fora dele, de modo que são necessárias $2i + 1$ comparações no total. Assim, o número médio de comparações usadas é igual a

$$\frac{3 + 5 + 7 + \cdots + (2n + 1)}{n} = \frac{2(1 + 2 + 3 + \cdots + n) + n}{n}.$$

Na Seção 4.1, mostraremos que

$$1 + 2 + 3 + \cdots + n = \frac{n(n + 1)}{2}.$$

Assim, o número médio de comparações usadas pelo algoritmo de busca linear (quando se sabe que x está na lista) é

$$\frac{2[n(n + 1)/2]}{n} + 1 = n + 2,$$

que é $\Theta(n)$. ◀

Lembre-se: Na análise do Exemplo 4 considerou-se que x estava na lista a ser pesquisada e que seria quase a mesma coisa se x estivesse em qualquer posição. Também é possível fazer uma análise do caso médio deste algoritmo quando x não estiver na lista (veja o Exercício 13 no final desta seção).

Lembre-se: Embora nós tenhamos contado as comparações necessárias para determinar se chegamos ao final do laço, elas não são geralmente contadas. A partir daqui, vamos ignorar tais comparações.

COMPLEXIDADE DE PIOR CASO DE DOIS ALGORITMOS DE ORDENAÇÃO Analisaremos a complexidade de pior caso do bubble sort e do insertion sort nos exemplos 5 e 6.

EXEMPLO 5 Qual é a complexidade de pior caso do bubble sort em termos do número de comparações feitas?

Solução: O bubble sort (descrito no Exemplo 4 da Seção 3.1) seleciona uma lista pela realização de uma seqüência de passos ao longo da lista. Durante cada passo, o bubble sort compara sucessivamente os elementos adjacentes, trocando-os se necessário. Quando o i-ésimo passo começa, os $i-1$ maiores elementos estão nas suas posições corretas. Durante esta etapa, $n-i$ comparações são usadas. Conseqüentemente, o número total de comparações usadas pelo bubble sort para ordenar um lista de n elementos é

$$(n-1)+(n-2)+\cdots+2+1=\frac{(n-1)n}{2}$$

usando-se uma fórmula de somatório que vamos demonstrar na Seção 4.1. Note que o bubble sort sempre usa essas comparações porque ela continua mesmo se a lista se tornar completamente ordenada em alguma etapa. Conseqüentemente, o bubble sort usa $(n-1)n/2$ comparações, assim ela tem uma complexidade de pior caso $\Theta(n^2)$ em termos do número de comparações usadas. ◀

EXEMPLO 6 Qual é a complexidade de pior caso do insertion sort em termos do número de comparações feitas?

Solução: O insertion sort (descrito na Seção 3.1) insere o j-ésimo elemento na ordem correta entre os primeiros $j-1$ elementos que já foram colocados na posição certa. Ele faz isso usando uma técnica de busca linear, comparando sucessivamente o j-ésimo elemento com o termo subseqüente até que um termo maior que ou igual àquele seja encontrado, ou compara a_j com ele mesmo e pára se a_j não for menor que ele mesmo. Conseqüentemente, no pior caso, j comparações são necessárias para inserir o j-ésimo elemento na posição correta. Assim, o número total de comparações usadas pelo insertion sort para organizar uma lista de n elementos é

$$2+3+\cdots+n=\frac{n(n+1)}{2}-1,$$

usando-se a fórmula de somatório para a soma de números inteiros consecutivos que demonstraremos na Seção 4.1, notando-se que o primeiro termo, 1, não está nesta soma. Note também que o insertion sort pode usar consideravelmente poucas comparações se os elementos menores aparecerem no final da lista. Concluímos que o insertion sort tem complexidade de pior caso $\Theta(n^2)$. ◀

Entendendo a Complexidade dos Algoritmos

A Tabela 1 apresenta a terminologia comumente usada para descrever a complexidade temporal dos algoritmos. Por exemplo, um algoritmo que encontra o maior termo dos primeiros 100 termos de uma lista de n elementos pela aplicação do Algoritmo 1 na seqüência dos 100 primeiros

TABELA 1 Terminologia Comumente Usada para a Complexidade de Algoritmos.

Complexidade	*Terminologia*
$\Theta(1)$	Complexidade constante
$\Theta(\log n)$	Complexidade logarítmica
$\Theta(n)$	Complexidade linear
$\Theta(n \log n)$	Complexidade $n \log n$
$\Theta(n^b)$	Complexidade polinomial
$\Theta(b^n)$, em que $b > 1$	Complexidade exponencial
$\Theta(n!)$	Complexidade fatorial

termos, em que n é um número inteiro com $n \geq 100$, tem **complexidade constante**, porque usa 99 comparações, não importando qual o valor de n (como o leitor pode verificar). O algoritmo de busca linear tem **complexidade linear** (pior caso ou caso médio) e o algoritmo de busca binária tem **complexidade logarítmica** (pior caso). Muitos algoritmos importantes têm complexidade $n \log n$, tal como a merge sort, que apresentaremos no Capítulo 4.

Um algoritmo tem **complexidade polinomial** se tiver complexidade $\Theta(n^b)$, em que b é um número inteiro com $b \geq 1$. Por exemplo, o algoritmo bubble sort é um algoritmo de tempo polinomial, pois usa $\Theta(n^2)$ comparações no pior caso. Um algoritmo tem **complexidade exponencial** se tiver uma complexidade temporal $\Theta(b^n)$, em que $b > 1$. O algoritmo que determina se uma proposição composta com n variáveis é satisfatória checando todas as possíveis tarefas que assumem algum valor-verdade é um algoritmo com complexidade exponencial, porque usa $\Theta(2^n)$ operações. Por fim, um algoritmo tem **complexidade fatorial** se tiver complexidade temporal $\Theta(n!)$. O algoritmo do caixeiro-viajante, que encontra todas as ordens em que um vendedor pode visitar n cidades, tem complexidade fatorial; vamos discutir este algoritmo no Capítulo 9.

Um problema que é possível resolver usando um algoritmo com complexidade polinomial de pior caso é chamado de **tratável**, porque a expectativa é de que ele fornecerá a solução para o problema de uma determinada entrada em um tempo relativamente curto. Entretanto, se o polinômio tiver estimativa Θ grande, que tem alto grau (como grau 100), ou se os coeficientes forem extremamente grandes, o algoritmo pode demorar muito para resolver o problema. Conseqüentemente, tal problema pode ser resolvido usando-se um algoritmo com complexidade temporal polinomial de pior caso, sem ter garantia de que o problema pode ser resolvido em um tempo razoável até mesmo para entradas relativamente pequenas. Felizmente, na prática, o grau e os coeficientes dos polinômios são geralmente pequenos.

A situação é muito pior para problemas que não podem ser resolvidos com um algoritmo com complexidade temporal polinomial de pior caso. Tais problemas são chamados de **intratáveis**. Geralmente, mas não sempre, uma grande quantidade de tempo é necessária para resolver o problema nos piores casos, até mesmo com valores de entrada pequenos. Na prática, entretanto, há situações em que um algoritmo com uma certa complexidade temporal de pior caso pode ser capaz de resolver um problema mais rápido para a maioria dos casos, com exceção da situação de pior caso. Quando queremos permitir que alguns casos, talvez poucos, não sejam resolvidos em um tempo razoável, a complexidade temporal de caso médio é uma medida melhor para saber quanto tempo um algoritmo leva para resolver um problema. Muitos problemas importantes na indústria estão próximos de serem intratáveis, mas podem ser praticamente resolvidos para essencialmente todos os conjuntos de entradas que aparecem no cotidiano. Outra maneira de lidar com problemas intratáveis, quando eles aparecem em aplicações práticas, é, em vez de procurar por soluções exatas de um problema, utilizar soluções aproximadas. Pode ser o caso de existir algoritmos rápidos para encontrar tais soluções aproximadas, talvez até com uma garantia de que elas não sejam muito diferentes de uma solução exata.

Existem alguns problemas em que não há algoritmo para resolvê-los. Tais problemas são conhecidos como **insolúveis** (em oposição a problemas **solúveis**, que podem ser resolvidos usando um algoritmo). A primeira demonstração de que existem problemas que não podem ser solucionados foi apresentada pelo grande matemático e cientista da computação, o inglês Alan Turing, quando ele mostrou que o problema da parada não pode ser resolvido. Lembre-se de que demonstramos que o problema da parada não pode ser resolvido na Seção 3.1. (A biografia de Alan Turing e a descrição de alguns de seus trabalhos podem ser encontradas no Capítulo 12.)

O estudo da complexidade dos algoritmos vai além do que descrevemos aqui. Note, entretanto, que muitos problemas passíveis de ser solucionados têm a propriedade de que nenhum algoritmo com complexidade temporal polinomial no pior caso pode resolvê-los, mas uma solução, se for conhecida, pode ser checada em um tempo polinomial. Os problemas cujas soluções podem ser verificadas em tempo polinomial pertencem à **classe NP** (os problemas tratáveis pertencem à **classe P**).* Há também uma importante classe de problemas, chamada de **problema NP-completo,** com a propriedade de que se qualquer um deles puder ser resolvido com um algoritmo com tempo polinomial no pior caso, então todos os problemas da classe NP podem ser resolvidos com algoritmos com tempo polinomial no pior caso.

* NP significa tempo *polinomial não-determinístico*.

O problema de satisfatibilidade é um exemplo de um problema NP-completo — podemos verificar rapidamente que um conjunto de valores-verdade para as variáveis de uma proposição composta torna-a verdadeira, mas nenhum algoritmo de tempo polinomial foi descoberto para encontrar tal conjunto de valores-verdade. [Por exemplo, uma busca exaustiva de todos os valores-verdade possíveis requer $\Theta(2^n)$ operações binárias, em que n é o número de variáveis na proposição composta.] Além disso, se um algoritmo com tempo polinomial para resolver um problema de satisfatibilidade for conhecido, haverá algoritmos com tempo polinomial para todos os problemas conhecidos nesta classe de problemas (e há muitos problemas importantes nesta classe).

Mesmo depois de uma busca extensiva, nenhum algoritmo com tempo polinomial no pior caso foi encontrado para qualquer problema desta classe. É aceitável, embora não demonstrado, que o problema NP-completo pode ser resolvido em tempo polinomial. Para mais informações sobre a complexidade dos algoritmos, consulte as referências, para esta seção, incluindo [CoLeRiSt01], listada no final deste livro. (Além disso, para mais discussões formais sobre complexidade computacional em termos de máquinas Turing, veja a Seção 12.5.)

Note que uma estimativa Θ grande da complexidade temporal de um algoritmo expressa como o tempo necessário para resolver o problema pode mudar com o crescimento das entradas. Na prática, é utilizada a melhor estimativa (ou seja, com a menor função referencial) que pode ser mostrada. Entretanto, Θ grande faz a estimativa da complexidade temporal, mas não pode ser diretamente transcrita no tempo do computador atualmente usado. Uma razão é que uma estimativa Θ grande de que $f(n)$ é $\Theta(g(n))$, em que $f(n)$ é a complexidade temporal de um algoritmo e $g(n)$ é a função referencial, significa que $C_1g(n) \leq f(n) \leq C_2g(n)$ quando $n > k$, em que C_1, C_2 e k são constantes. Então, sem conhecer as constantes C_1, C_2 e k na inequação, essa estimativa não pode ser usada para determinar uma limitante superior e uma inferior em relação ao número de operações usadas no pior caso. Como foi dito anteriormente, o tempo necessário para uma operação depende do tipo de operação e do computador a ser usados. Geralmente, em vez de uma estimativa Θ grande da complexidade temporal de pior caso de um algoritmo, temos apenas uma estimativa O grande. Note que uma estimativa O grande da complexidade temporal de um algoritmo fornece uma limitante superior, mas não uma inferior, do tempo de pior caso necessário para o algoritmo como uma função de determinado tamanho. No entanto, para simplificar, usaremos normalmente estimativas O grande ao descrever a complexidade temporal dos algoritmos, a partir do entendimento de que a estimativa Θ grande pode fornecer mais informações.

Entretanto, o tempo necessário para um algoritmo resolver um problema de tamanho específico pode ser determinado se todas as operações puderem ser reduzidas às operações binárias usadas por um computador. A Tabela 2 apresenta o tempo necessário para resolver problemas de vários tamanhos com um algoritmo usando o número indicado de operações binárias. Tempos com mais de 10^{100} anos são indicados com um asterisco. (Na Seção 3.6 será discutido o número de operações binárias usado para adicionar e multiplicar dois números inteiros.) Na construção dessa tabela, considera-se que cada operação binária leva 10^{-9} segundos, que é o tempo necessário para uma operação binária que utilize os computadores mais rápidos da atualidade. No futuro, esses tempos tendem a diminuir à medida que computadores mais rápidos forem sendo desenvolvidos.

TABELA 2 O Tempo de Computação Usado pelos Algoritmos.

Tamanho do Problema	*Operações binárias utilizadas*					
n	$\log n$	n	$n \log n$	n^2	2^n	$n!$
10	3×10^{-9} s	10^{-8} s	3×10^{-8} s	10^{-7} s	10^{-6} s	3×10^{-3} s
10^2	7×10^{-9} s	10^{-7} s	7×10^{-7} s	10^{-5} s	4×10^{13} anos	*
10^3	$1{,}0 \times 10^{-8}$ s	10^{-6} s	1×10^{-5} s	10^{-3} s	*	*
10^4	$1{,}3 \times 10^{-8}$ s	10^{-5} s	1×10^{-4} s	10^{-1} s	*	*
10^5	$1{,}7 \times 10^{-8}$ s	10^{-4} s	2×10^{-3} s	10 s	*	*
10^6	2×10^{-8} s	10^{-3} s	2×10^{-2} s	17 min	*	*

É importante saber quanto tempo um computador precisa para resolver um problema. Por exemplo, se um algoritmo necessita de 10 horas, pode ser melhor gastar o tempo do computador (e dinheiro) para resolver este problema. Mas, se um algoritmo necessita de 10 bilhões de anos para resolver um problema, seria irracional usar recursos para implementá-lo. Um dos fenômenos mais interessantes da tecnologia moderna é o grande aumento da velocidade e da memória dos computadores. Outro fator importante que diminui o tempo necessário para resolver problemas em computadores é o **processamento paralelo,** que é a técnica de realizar seqüências de operações simultaneamente.

Algoritmos eficientes, incluindo muitos algoritmos com complexidade temporal polinomial, promovem melhorias significativas da tecnologia. Entretanto, essas melhorias tecnológicas oferecem uma pequena ajuda na superação da complexidade temporal de algoritmos exponencial ou fatorial. Com o crescimento da velocidade de computação, o aumento da memória do computador e o uso dos algoritmos que tiram vantagem do processamento paralelo, muitos problemas que eram considerados impossíveis de ser solucionados cinco anos atrás são agora resolvidos rapidamente, e certamente daqui a cinco anos esta proposição continuará sendo verdadeira.

Exercícios

1. Quantas comparações são usadas pelo algoritmo dado no Exercício 16 da Seção 3.1 para encontrar o menor número natural em uma seqüência de n números naturais?
2. Escreva o algoritmo que coloca os primeiros quatro termos de uma lista de comprimento arbitrário em ordem crescente. Mostre que este algoritmo tem complexidade temporal $O(1)$ em termos do número de comparações usadas.
3. Suponha que um elemento é conhecido por estar entre os quatro primeiros elementos de uma lista com 32 elementos. Uma busca linear ou uma busca binária localizaria este elemento mais rapidamente?
4. Determine o número de multiplicações usadas para encontrar x^{2^k} começando com x e elevando ao quadrado sucessivamente (para encontrar x^2, x^4, e assim por diante). Esta é a maneira mais eficiente de encontrar x^{2^k} em vez de multiplicar x por ele mesmo quantas vezes forem necessárias?
5. Dê a estimativa O grande para o número de comparações usadas pelo algoritmo que determina o número de 1s em uma cadeia de bits examinando cada bit da cadeia para determinar se é um bit 1 (veja o Exercício 25 da Seção 3.1).

*6. **a)** Mostre que este algoritmo determina o número de bits 1 na cadeia S:

```
procedure contagem de bit (S: cadeia de bits)
contagem := 0
while S ≠ 0
begin
  contagem := contagem + 1
  S := S ∧ (S − 1)
end {contagem é o número de 1s em S}
```

Aqui, $S - 1$ é a cadeia de bits obtida trocando-se o bit 1 mais à direita de S por um 0 e todos os bits 0 à direita deste por 1s. [Lembre-se de que $S \wedge (S - 1)$ é o operador *AND* de S e $S - 1$.]

b) Quantas operações com operadores *AND* são necessárias para encontrar o número de bits 1 em uma cadeia S?

7. O algoritmo convencional para avaliar um polinômio $a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ em $x = c$ pode ser expresso em pseudocódigo por

```
procedure polinomio(c, a0, a1, . . . , an: real)
potencia := 1
y := a0
for i := 1 to n
begin
  potencia := potencia * c
  y := y + ai * potencia
end {y = anc^n + an−1c^n−1 + · · · + a1c + a0}
```

em que o valor final de y é o valor do polinômio em $x = c$.

a) Avalie $3x^2 + x + 1$ com $x = 2$ trabalhando cada passo do algoritmo, mostrando os valores obtidos em cada passo.

b) Exatamente quantas multiplicações e adições são usadas para avaliar um polinômio de grau n com $x = c$? (Não conte as adições usadas para incrementar a variável do laço.)

8. Há um algoritmo mais eficiente (em termos do número de multiplicações e adições usadas) para avaliar polinômios em vez do algoritmo convencional descrito no exercício anterior. É chamado de **método de Horner.** Este pseudocódigo mostra como usar este método para encontrar o valor de $a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ em $x = c$.

```
procedure Horner(c, a0, a1, a2, . . . , an: real)
y := an
for i := 1 to n
  y := y * c + an−i
{y = anc^n + an−1c^n−1 + · · · + a1c + a0}
```

a) Avalie $3x^2 + x + 1$ com $x = 2$ trabalhando cada passo do algoritmo, mostrando os valores obtidos em cada passo.

b) Exatamente quantas multiplicações e adições são usadas por este algoritmo para avaliar um polinômio de grau n com $x = c$? (Não conte as adições usadas para incrementar a variável do laço.)

9. Qual o maior valor de n para que se possa resolver em um segundo um problema que usa um algoritmo que necessita de $f(n)$ operações binárias, em que cada operação é realizada em 10^{-9} segundos, com os seguintes valores para $f(n)$?
a) $\log n$ **b)** n **c)** $n \log n$
d) n^2 **e)** 2^n **f)** $n!$

10. Quanto tempo um algoritmo leva para resolver um problema de tamanho n se ele usar $2n^2 + 2^n$ operações binárias, cada uma demorando 10^{-9} segundos, com os seguintes valores para n?
a) 10 **b)** 20 **c)** 50 **d)** 100

11. De quanto tempo um algoritmo que usa 2^{50} operações binárias necessita, se cada operação leva os tempos abaixo?
a) 10^{-6} segundos **b)** 10^{-9} segundos **c)** 10^{-12} segundos

12. Determine o menor número de comparações, ou a melhor performance
a) necessária para encontrar o valor máximo de uma seqüência de n números inteiros, usando o Algoritmo 1 da Seção 3.1.
b) usada para localizar um elemento de uma lista com n termos com a busca linear.
c) usada para localizar um elemento de uma lista com n termos com a busca binária.

13. Analise a performance de caso médio do algoritmo de busca linear se em metade do tempo o elemento x não está na lista e x estar na lista significa estar em qualquer posição.

14. Um algoritmo é chamado de **ótimo** para a solução de um problema de uma determinada operação se não houver algoritmo para resolver este problema usando menos operações.
a) Mostre que o Algoritmo 1 da Seção 3.1 é um algoritmo ótimo com relação ao número de comparações dos números inteiros. (*Nota:* As comparações usadas para contar o agrupamento não estão em questão aqui.)
b) O algoritmo de busca linear é ótimo com relação ao número de comparações de números inteiros (não incluindo as comparações usadas para contar o agrupamento)?

15. Descreva a complexidade temporal de pior caso, medida em termos de comparações, do algoritmo de busca ternária descrito no Exercício 27 da Seção 3.1.

16. Descreva a complexidade temporal de pior caso, medida em termos de comparações, do algoritmo de busca descrito no Exercício 28 da Seção 3.1.

17. Analise a complexidade temporal de pior caso do algoritmo que você construiu no Exercício 29 da Seção 3.1 para localizar uma moda em uma lista de números inteiros não decrescentes.

18. Analise a complexidade temporal de pior caso do algoritmo que você construiu no Exercício 30 da Seção 3.1 para localizar todas as modas em uma lista de números inteiros não decrescentes.

19. Analise a complexidade temporal de pior caso do algoritmo que você construiu no Exercício 31 da Seção 3.1 para encontrar o primeiro termo da seqüência de números inteiros igual a algum termo anterior.

20. Analise a complexidade temporal de pior caso do algoritmo que você construiu no Exercício 32 da Seção 3.1 para encontrar todos os termos de uma seqüência que sejam maiores que a soma de todos os elementos anteriores.

21. Analise a complexidade temporal de pior caso do algoritmo que você construiu no Exercício 33 da Seção 3.1 para encontrar o primeiro termo de uma seqüência que seja menor que o termo imediatamente anterior.

22. Determine a complexidade de pior caso em termos de comparações do algoritmo do Exercício 5 da Seção 3.1 para determinar todos os valores que aparecem mais de uma vez em uma lista ordenada de números inteiros.

23. Determine a complexidade de pior caso em termos de comparações do algoritmo do Exercício 9 da Seção 3.1 para determinar se uma cadeia é um palíndromo.

24. Quantas comparações o selection sort (veja o preâmbulo do Exercício 41 da Seção 3.1) usa para ordenar n itens? Use sua resposta para dar uma estimativa O grande da complexidade do selection sort em termos de número de comparações para o selection sort.

25. Encontre uma estimativa O grande para a complexidade de pior caso em termos do número de comparações usadas e o número de termos trocados pelo insertion sort binário descrita no preâmbulo do Exercício 47 da Seção 3.1.

26. Mostre que o algoritmo voraz para organizar trocos de n centavos usando moedas de 25, 10, 5 e 1 centavo tem complexidade $O(n)$, medida em termos de comparações necessárias.

27. Descreva como o número de comparações usadas no pior caso muda quando os algoritmos abaixo são usados para a procura de um elemento em uma lista quando o tamanho da lista dobra de n para $2n$, em que n é um número inteiro positivo.
a) busca linear **b)** busca binária

28. Descreva como o número de comparações usadas no pior caso muda quando o tamanho da lista a ser ordenada dobra de n para $2n$, em que n é um número inteiro positivo, quando os algoritmos de ordenação abaixo são usados.
a) bubble sort **b)** insertion sort
c) selection sort (descrito no preâmbulo do Exercício 41 da Seção 3.1)
d) insertion sort binário (descrito no preâmbulo do Exercício 47 da Seção 3.1)

3.4 Os Números Inteiros e a Divisão

Introdução

A parte da matemática que envolve os números inteiros e suas propriedades pertence ao ramo da matemática chamado de **teoria dos números.** Esta seção é o início de uma introdução composta de

quatro secções da teoria dos números. Vamos desenvolver os conceitos básicos da teoria dos números usados na ciência da computação. Nesta seção, recordaremos os conceitos básicos da teoria dos números, incluindo divisibilidade e aritmética modular. Na Seção 3.5, estudaremos os números primos e começaremos nossa discussão sobre os máximos divisores comuns. Na Seção 3.6, descreveremos diversos algoritmos importantes da teoria dos números, relacionando o conteúdo das Seções 3.1 e 3.3, sobre algoritmos e sua complexidade, com as noções introduzidas nesta seção e na Seção 3.5. Por exemplo, introduziremos os algoritmos que permitem encontrar o máximo divisor comum de dois números inteiros positivos e realizar a aritmética computacional usando expansões binárias. Por fim, na Seção 3.7, continuaremos nosso estudo sobre teoria dos números introduzindo alguns resultados importantes e suas aplicações para a aritmética computacional e a criptografia, o estudo das mensagens secretas. Contaremos com os métodos de demonstrações estudados no Capítulo 1 para desenvolver os teoremas básicos da teoria dos números. Isto lhe dará oportunidade de usar o que aprendeu sobre métodos e estratégias de demonstrações em uma nova construção.

As idéias que vamos desenvolver nesta seção são baseadas na noção de divisibilidade. A divisão de um número inteiro por um número inteiro positivo produz um quociente e um resto. Trabalhar com esses restos conduz à aritmética modular, que é usada pela ciência da computação. Serão discutidas três aplicações de aritmética modular nesta seção: construção de números pseudo-aleatórios, determinação de localização da memória para arquivos de um computador e codificação e decodificação de mensagens.

Divisão

Quando um número inteiro é dividido por um segundo número inteiro, diferente de zero, o quociente pode ou não ser um número inteiro. Por exemplo, $12/3 = 4$ é um número inteiro, em contraposição $11/4 = 2{,}75$ não é. Isto nos leva à Definição 1.

DEFINIÇÃO 1 Se a e b são números inteiros com $a \neq 0$, dizemos que *a divide b* se houver um número inteiro c de modo que $b = ac$. Quando a divide b, dizemos que a é um *fator* de b e que b é um *múltiplo* de a. A notação $a \mid b$ indica que a divide b. Escrevemos $a \nmid b$ quando a não divide b.

Lembre-se: Podemos expressar $a \mid b$ usando quantificadores como $\exists c(ac = b)$, em que o universo do discurso é o conjunto dos números inteiros.

Na Figura 1, uma linha numérica indica quais números inteiros são divisíveis pelo número inteiro positivo d.

EXEMPLO 1 Determine se $3 \mid 7$ e se $3 \mid 12$.

Solução: Temos que $3 \nmid 7$, pois $7/3$ não é um número inteiro. Por outro lado, $3 \mid 12$, pois $12/3 = 4$. ◀

EXEMPLO 2 Considere n e d como números inteiros positivos. Quantos números inteiros positivos, não excedentes a n, são divisíveis por d?

Solução: Os números inteiros positivos divisíveis por d são todos os números inteiros na forma dk, em que k é um número inteiro positivo. Assim, o número de inteiros positivos divisíveis por d que não excedem a n é igual ao número de inteiros k com $0 < dk \leq n$, ou com $0 < k \leq n/d$. Então, há $\lfloor n/d \rfloor$ números inteiros positivos não excedentes a n que são divisíveis por d. ◀

Exemplos Extras

Algumas das propriedades básicas de divisibilidade de números inteiros são dadas pelo Teorema 1.

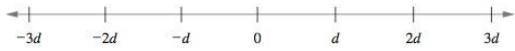


FIGURA 1 Números Inteiros Divisíveis pelo Número Inteiro Positivo *d*.

TEOREMA 1 Sejam a, b e c números inteiros. Então,

(i) se $a \mid b$ e $a \mid c$, então $a \mid (b + c)$;
(ii) se $a \mid b$, então $a \mid bc$ para todo número inteiro c;
(iii) se $a \mid b$ e $b \mid c$, então $a \mid c$.

Demonstração: Daremos uma demonstração direta de (i). Suponha que $a \mid b$ e $a \mid c$. Então, a partir da definição de divisibilidade, temos que há números inteiros s e t com $b = as$ e $c = at$. Assim,

$$b + c = as + at = a(s + t).$$

Então, a divide $b + c$. Isto estabelece o item (i) do teorema. As demonstrações dos itens (ii) e (iii) são deixadas como exercício para o leitor. ◁

O Teorema 1 tem esta conseqüência útil.

COROLÁRIO 1 Se a, b e c são números inteiros, tal que $a \mid b$ e $a \mid c$, então $a \mid mb + nc$ sempre que m e n forem números inteiros.

Demonstração: Daremos uma demonstração direta. A partir do item (ii) do Teorema 1, temos que $a \mid mb$ e $a \mid nc$ sempre que m e n forem números inteiros. A partir do item (i) do Teorema 1, temos que $a \mid mb + nc$. ◁

O Algoritmo de Divisão

Quando um número inteiro é dividido por um número inteiro positivo, há um quociente e um resto, como o algoritmo de divisão mostra.

TEOREMA 2 **O ALGORITMO DE DIVISÃO** Considere a como um número inteiro e d como um número inteiro positivo. Então, haverá números inteiros q e r, únicos, com $0 \leq r < d$, tal que $a = dq + r$.

Apresentamos a demonstração do algoritmo de divisão no Exemplo 5 e no Exercício 37 da Seção 4.2.

Lembre-se: O Teorema 2 não é realmente um algoritmo. (Por que não?) No entanto, usamos esta nomenclatura tradicional.

DEFINIÇÃO 2 Na equação dada no algoritmo de divisão, d é chamado de *divisor*, a é chamado de *dividendo*, q é chamado de *quociente* e r é chamado de *resto*. Esta notação é usada para expressar o quociente e o resto:

$$q = a \textbf{ div } d, \qquad r = a \textbf{ mod } d.$$

Os exemplos 3 e 4 ilustram o algoritmo de divisão.

EXEMPLO 3 Quais são o quociente e o resto quando 101 é dividido por 11?

Solução: Temos

$$101 = 11 \cdot 9 + 2.$$

Assim, o quociente quando 101 é dividido por 11 é $9 = 101$ **div** 11, e o resto é $2 = 101$ **mod** 11. ◀

EXEMPLO 4 Quais são o quociente e o resto quando -11 é dividido por 3?

Solução: Temos

$$-11 = 3(-4) + 1.$$

Exemplos Extras

Assim, quando -11 é dividido por 3, o quociente é $-4 = -11$ **div** 3, e o resto é $1 = -11$ **mod** 3.

Note que o resto não pode ser negativo. Conseqüentemente, o resto *não* é -2, mesmo se

$$-11 = 3(-3) - 2,$$

porque $r = -2$ não satisfaz $0 \leq r < 3$. ◀

Note que o número inteiro a é divisível pelo número inteiro d se e somente se o resto for zero quando a é dividido por d.

Aritmética Modular

Em algumas situações, nos preocupamos apenas com o resto de um número inteiro quando ele é dividido por um determinado número inteiro positivo. Por exemplo, quando perguntamos que horas serão daqui a 50 horas (em um relógio de 24 horas), nos preocupamos apenas com o resto quando 50 mais a hora atual é dividido por 24. Como estamos interessados apenas nos restos, temos uma notação especial para eles.

Temos uma notação para indicar que dois números inteiros têm o mesmo resto quando eles são divididos pelo número inteiro positivo m.

DEFINIÇÃO 3 Se a e b forem números inteiros e m for um número inteiro positivo, então a é *congruente a* b *módulo* m se m divide $a - b$. Usamos a notação $a \equiv b \pmod{m}$ para indicar que a é congruente a b módulo m. Se a e b não forem congruentes módulo m, escrevemos $a \not\equiv b \pmod{m}$.

A conexão entre as notações usadas quando trabalhamos com restos torna-se clara com o Teorema 3.

TEOREMA 3 Considere a e b como números inteiros e considere m como um número inteiro positivo. Então, $a \equiv b \pmod{m}$ se e somente se a **mod** $m = b$ **mod** m.

A demonstração do Teorema 3 será apresentada nos exercícios 11 e 12, no final desta seção.

EXEMPLO 5 Determine se 17 é congruente a 5 módulo 6 e se 24 e 14 são congruentes ao módulo 6.

Solução: Como 6 divide $17 - 5 = 12$, vemos que $17 \equiv 5 \pmod 6$. Entretanto, como $24 - 14 = 10$ não é divisível por 6, temos que $24 \not\equiv 14 \pmod 6$. ◀

O grande matemático alemão Karl Friedrich Gauss desenvolveu o conceito de congruência no final do século XVIII.

A noção de congruência desempenha um importante papel no desenvolvimento da teoria dos números. O Teorema 4 fornece uma maneira útil de trabalhar com congruências.

TEOREMA 4 Considere m como um número inteiro positivo. Os números inteiros a e b são congruentes ao módulo m se e somente se houver um número inteiro k, tal que $a = b + km$.

Demonstração: Se $a \equiv b \pmod m$, então $m \mid (a - b)$. Isto significa que há um número inteiro k tal que $a - b = km$, para que $a = b + km$. Reciprocamente, se houver um número inteiro k, tal que $a = b + km$, então $km = a - b$. Assim, m divide $a - b$, e portanto $a \equiv b$ (mód. m). ◁

O conjunto de todos os números inteiros congruentes a um inteiro a módulo m é chamado de **classe de congruência** de a módulo m. No Capítulo 8 vamos mostrar que há m conjuntos disjuntos de classes de equivalência módulo m e que a união dessas classes de equivalência é o conjunto dos números inteiros.

O Teorema 5 mostra como a congruência funciona com relação à adição e à multiplicação.

TEOREMA 5 Considere m como um número inteiro positivo. Se $a \equiv b \pmod m$ e $c \equiv d \pmod m$, então
$a + c \equiv b + d \pmod m$ e $ac \equiv bd \pmod m$.

Demonstração: Como $a \equiv b \pmod m$ e $c \equiv d \pmod m$, há números inteiros s e t com $b = a + sm$ e $d = c + tm$. Assim,

$$b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$$

e

$$bd = (a + sm)(c + tm) = ac + m(at + cs + stm).$$

Assim,

$$a + c \equiv b + d \pmod m \quad \text{e} \quad ac \equiv bd \pmod m.$$ ◁

Links

KARL FRIEDRICH GAUSS (1777–1855) Karl Friedrich Gauss, filho de um pedreiro, foi uma criança-prodígio. Ele demonstrou seu potencial com 10 anos de idade, quando rapidamente resolveu um problema dado pelo professor para manter a sala em silêncio. O professor pediu aos estudantes que encontrassem a soma dos primeiros 100 números inteiros positivos. Gauss viu que a soma poderia ser encontrada formando 50 pares, cada um com a soma 101: $1 + 100$, $2 + 99, \ldots, 50 + 51$. Este brilhantismo atraiu o patrocínio de benfeitores, incluindo o Duque Ferdinand de Brunswick, que tornou possível a Gauss freqüentar a Caroline College e a Universidade de Göttingen. Enquanto estudante, ele inventou o método dos mínimos quadrados, que é usado para estimar o valor mais verossímil de uma variável a partir de resultados experimentais. Em 1796, Gauss fez uma descoberta fundamental na geometria, desenvolvendo um assunto que não era discutido desde os tempos remotos. Ele mostrou que um polígono regular com 17 lados poderia ser construído usando-se apenas uma régua e um compasso. Em 1799, Gauss apresentou a primeira regra rigorosa do Teorema Fundamental de Álgebra, que estabelece que um polinômio de grau n tem exatamente n raízes (contando multiplicidades). Ele alcançou fama mundial quando calculou corretamente a órbita do primeiro asteróide descoberto, Ceres, usando dados limitados.

Gauss foi chamado de Príncipe da Matemática pelos seus matemáticos contemporâneos. Embora seja conhecido por suas descobertas em geometria, álgebra, análise, astronomia e física, ele tinha um interesse especial na teoria dos números, o que pode ser visto na declaração: "A matemática é a rainha das ciências, e a teoria dos números, a rainha da matemática". Gauss deixou os fundamentos para a teoria dos números moderna com a publicação de seu livro, *Disquisitiones Arithmeticae,* em 1801.

EXEMPLO 6 Como $7 \equiv 2 \pmod 5$ e $11 \equiv 1 \pmod 5$, temos que, a partir do Teorema 5,

$$18 = 7 + 11 \equiv 2 + 1 = 3 \pmod 5$$

e que

$$77 = 7 \cdot 11 \equiv 2 \cdot 1 = 2 \pmod 5.$$ ◀

Precisaremos do corolário a seguir do Teorema 5 da Seção 4.4.

COROLÁRIO 2 Considere m como um número inteiro positivo e a e b como números inteiros. Então

$$(a + b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m$$

e

$$ab \bmod m = ((a \bmod m)(b \bmod m)) \bmod m.$$

Demonstração: Pelas definições de **mod** m e pela definição de congruência módulo m, sabemos que $a \equiv (a \bmod m) \pmod m$ e $b \equiv (b \bmod m) \pmod m$. Assim, o Teorema 5 nos diz que

$$a + b \equiv (a \bmod m) + (b \bmod m) \pmod m$$

e

$$ab \equiv (a \bmod m)(b \bmod m) \pmod m.$$

As equações deste corolário seguem a partir das duas últimas congruências do Teorema 3. ◁

Devemos ter cuidado ao trabalhar com as congruências. Algumas propriedades que pensamos ser verdadeiras não são válidas. Por exemplo, se $ac \equiv bc \pmod m$, a congruência $a \equiv b \pmod m$ pode ser falsa. Da mesma forma, se $a \equiv b \pmod m$ e $c \equiv d \pmod m$, a congruência $a^c \equiv b^d \pmod m$ pode também ser falsa. (Veja o Exercício 23.)

Aplicações das Congruências

A teoria dos números tem aplicações em diversas áreas. Introduziremos três aplicações nesta seção: o uso de congruências para determinar a localização da memória de arquivos de computador, a construção de números pseudo-aleatórios e a codificação de sistemas com base na aritmética modular.

EXEMPLO 7 **Funções de Hashing** O computador central em uma companhia de seguros administra os registros para cada um de seus clientes. Como a localização da memória pode ser determinada para que os registros dos clientes sejam recuperados rapidamente? A solução para este problema é usar uma escolha apropriada: **a função de hashing.** Os registros são identificados usando-se uma **chave,** que identifica unicamente cada registro do cliente. Por exemplo, os registros de cliente são geralmente identificados usando-se um número de Segurança Social do cliente como a chave. Uma função de hashing h determina a localização da memória $h(k)$ para o registro que tem k como sua chave.

Links

Na prática, muitas funções de hashing diferentes são usadas. Uma das mais comuns é a função

$$h(k) = k \bmod m$$

em que m é o número de localizações da memória disponíveis.

As funções de hashing podem ser facilmente avaliadas para que os arquivos possam ser encontrados rapidamente. A função de hashing $h(k) = k \bmod m$ tem esta característica; para encontrar $h(k)$, precisamos apenas computar o resto quando k é dividido por m. Além disso, a função de hashing deve ser sobrejetiva para que todas as localizações de memória sejam utilizadas. A função $h(k) = k \bmod m$ também satisfaz esta propriedade.

Por exemplo, quando $m = 111$, o registro do cliente com o número de Seguro Social 064212848 é projetado para a localização da memória 14, porque

$$h(064212848) = 064212848 \bmod 111 = 14.$$

Da mesma forma, como

$$h(037149212) = 037149212 \bmod 111 = 65,$$

o registro do cliente com o número de Seguro Social 037149212 é projetado para a localização da memória 65.

Como a função de hashing não é injetora (porque há mais chaves possíveis que localizações da memória), mais de um arquivo pode ser projetado para a localização. Quando isso acontece, dizemos que ocorreu uma **colisão**. Uma maneira de resolver uma colisão é determinar o primeiro local vago seguido daquele já ocupado que foi projetado pela função de hashing. Por exemplo, depois de designar duas tarefas anteriores, determinamos a localização 15 para o registro do cliente com o número de Seguro Social 107405723. Para verificar isto, primeiro note que $h(k)$ mapeia este número para a localização 14, porque

$$h(107405723) = 107405723 \bmod 111 = 14,$$

mas este local já está ocupado (pelo arquivo do cliente com o número de Seguro Social 064212848). Entretanto, a localização 15, o primeiro local subseqüente à localização 14, está livre.

Há muitas maneiras mais sofisticadas para resolver colisões que são mais eficientes que este simples método que descrevemos. Elas serão discutidas nas referências às funções de hashing dadas no final deste livro. ◀

EXEMPLO 8 **Números Pseudo-aleatórios** Números escolhidos aleatoriamente são geralmente necessários em simulações de computação. Diferentes métodos foram desenvolvidos para gerar números que têm propriedades de escolher aleatoriamente números. Como os números gerados por métodos sistêmicos não são verdadeiramente aleatórios, eles são chamados de **números pseudo-aleatórios.**

O procedimento mais comum usado para gerar esses números é o **método congruente linear.** Escolhemos quatro números inteiros: o **módulo** m, o **multiplicador** a, o **acréscimo** c e a **origem** x_0, com $2 \leq a < m$, $0 \leq c < m$ e $0 \leq x_0 < m$. Construímos uma seqüência de números pseudo-aleatórios $\{x_n\}$, com $0 \leq x_n < m$ para todo n, usando sucessivamente a congruência

$$x_{n+1} = (ax_n + c) \bmod m.$$

(Este é um exemplo de definição recursiva, a ser discutida na Seção 4.3, quando mostraremos que tais seqüências são bem definidas.)

Muitas experiências com computadores necessitam de números pseudo-aleatórios entre 0 e 1. Para gerá-los, dividimos os números gerados com um gerador de congruência linear pelo módulo, ou seja, usamos os números x_n/m.

Por exemplo, a seqüência de números pseudo-aleatórios gerados escolhendo-se $m = 9$, $a = 7$, $c = 4$ e $x_0 = 3$, pode ser encontrada a seguir:

$x_1 = 7x_0 + 4 \bmod 9 = 7 \cdot 3 + 4 \bmod 9 = 25 \bmod 9 = 7$
$x_2 = 7x_1 + 4 \bmod 9 = 7 \cdot 7 + 4 \bmod 9 = 53 \bmod 9 = 8$
$x_3 = 7x_2 + 4 \bmod 9 = 7 \cdot 8 + 4 \bmod 9 = 60 \bmod 9 = 6$
$x_4 = 7x_3 + 4 \bmod 9 = 7 \cdot 6 + 4 \bmod 9 = 46 \bmod 9 = 1$
$x_5 = 7x_4 + 4 \bmod 9 = 7 \cdot 1 + 4 \bmod 9 = 11 \bmod 9 = 2$
$x_6 = 7x_5 + 4 \bmod 9 = 7 \cdot 2 + 4 \bmod 9 = 18 \bmod 9 = 0$
$x_7 = 7x_6 + 4 \bmod 9 = 7 \cdot 0 + 4 \bmod 9 = 4 \bmod 9 = 4$
$x_8 = 7x_7 + 4 \bmod 9 = 7 \cdot 4 + 4 \bmod 9 = 32 \bmod 9 = 5$
$x_9 = 7x_8 + 4 \bmod 9 = 7 \cdot 5 + 4 \bmod 9 = 39 \bmod 9 = 3$

Como $x_9 = x_0$ e cada termo depende apenas de seu termo anterior, a seqüência abaixo é gerada:

3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, ...

Esta seqüência contém nove números diferentes antes de qualquer repetição.

A maioria dos computadores usa o gerador de congruência linear para gerar números pseudo-aleatórios. Geralmente, um gerador de congruência linear com acréscimo $c = 0$ é usado. Esse gerador é chamado de **gerador multiplicativo puro.** Por exemplo, o gerador multiplicativo puro com módulo $2^{31} - 1$ e multiplicador $7^5 = 16\,807$ é muito usado. Com esses valores, podemos mostrar que $2^{31} - 2$ números são gerados antes de qualquer repetição. ◀

Criptologia

As congruências têm muitas aplicações em matemática discreta e ciência da computação. Discussões sobre essas aplicações podem ser encontradas nas leituras sugeridas no final do livro. Uma das aplicações mais importantes da congruência envolve a **criptologia,** que é o estudo das mensagens secretas. Um dos usos mais antigos conhecido da criptologia foi com Júlio César. Ele fez mensagens secretas substituindo cada letra pelas três letras à frente no alfabeto (sendo as últimas três letras do alfabeto substituídas pelas três primeiras). Por exemplo, usando este esquema, a letra *B* é enviada como *E* e a letra *X*, como *A*. Este é um exemplo de **codificação,** ou seja, o processo de montar uma mensagem secreta.

Para expressar matematicamente o processo de codificação de César, primeiro substitua cada letra por um número inteiro de 0 a 25, com base em sua posição no alfabeto. Por exemplo, substitua *A* por 0, *K* por 10 e *Z* por 25. O método de codificação de César pode ser representado pela função f, que projeta para o número inteiro não negativo p, $p \leq 25$, o inteiro $f(p)$ do conjunto $\{0, 1, 2, \ldots, 25\}$ com

$$f(p) = (p + 3) \bmod 26.$$

Na versão codificada da mensagem, a letra representada por p é substituída pela letra representada por $(p + 3) \bmod 26$.

EXEMPLO 9 Qual é a mensagem secreta produzida a partir da mensagem "MEET YOU IN THE PARK" usando a criptografia César?

Solução: Primeiro troque as letras da mensagem por números. Isto produz

12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10.

Agora substitua cada um desses números p por $f(p) = (p + 3)$ **mod** 26. Isto equivale a

15 7 7 22 1 17 23 11 16 22 10 7 18 3 20 13.

Transcrevendo de volta às letras, temos a mensagem codificada "PHHW BRX LQ WKH SDUN." ◀

Para recuperar a mensagem original a partir de uma mensagem secreta codificada pelo código de César, a função f^{-1}, a inversa de f, é usada. Note que a função f^{-1} projeta um número inteiro p de $\{0, 1, 2, \ldots, 25\}$ para $f^{-1}(p) = (p - 3)$ **mod** 26. Em outras palavras, para encontrar a mensagem original, cada letra é substituída pelas três letras anteriores no alfabeto, com as primeiras três letras referindo-se às três últimas letras do alfabeto. O processo de determinar a mensagem original a partir da mensagem codificada é chamado de **decodificação.**

Há várias maneiras de construir o código de César. Por exemplo, em vez de substituir cada letra por uma que esteja 3 posições à frente, podemos substituir cada letra por k letras à frente, de tal forma que

$$f(p) = (p + k) \textbf{ mod } 26.$$

Esse código é chamado de **código de substituição.** Note que a decodificação pode ser realizada usando-se

$$f^{-1}(p) = (p - k) \textbf{ mod } 26.$$

É óbvio que o método de César e os códigos de substituição não fornecem um alto nível de segurança. Há várias maneiras de melhorar este método. Uma aproximação que aumenta um pouco a segurança é usar a função de forma

$$f(p) = (ap + b) \textbf{ mod } 26,$$

em que a e b são números inteiros, escolhidos para que f seja uma bijetora. (Tal função é chamada de *transformação afim*.) Isto fornece diversos sistemas de codificações possíveis. O uso de um desses sistemas é ilustrado no Exemplo 10.

EXEMPLO 10 Qual letra substitui a letra K quando a função $f(p) = (7p + 3)$ **mod** 26 é usada para codificação?

Solução: Primeiro, note que 10 representa K. Então, usando a função de codificação específica, temos que $f(10) = (7 \cdot 10 + 3)$ **mod** $26 = 21$. Como 21 representa V, K é substituída por V na mensagem codificada. ◀

Links O método de codificação de César e sua construção funcionam pela substituição de cada letra do alfabeto por outra. Métodos de codificação desse tipo são vulneráveis a ataques baseados na freqüência de ocorrência das letras na mensagem. Métodos de codificação mais sofisticados são baseados na substituição de blocos de letras por outros blocos de letras. Há várias técnicas baseadas em aritmética modular para codificar blocos de letras. Outras discussões sobre o assunto podem ser encontradas nas leituras sugeridas no fim deste livro.

Exercícios

1. O número 17 divide cada um dos números abaixo?
   a) 68 **b)** 84 **c)** 357 **d)** 1001
2. Mostre que se a for um número inteiro diferente de 0, então
   a) 1 divide a. **b)** a divide 0.
3. Mostre que o item (*ii*) do Teorema 1 é verdadeiro.
4. Mostre que o item (*iii*) do Teorema 1 é verdadeiro.
5. Mostre que se $a \mid b$ e $b \mid a$, em que a e b são números inteiros, então $a = b$ ou $a = -b$.
6. Mostre que se a, b, c e d são números inteiros, tal que $a \mid c$ e $b \mid d$, então $ab \mid cd$.

7. Mostre que se a, b e c são números inteiros com $c \neq 0$, tal que $ac \mid bc$, então $a \mid b$.

8. Demonstre ou negue que se $a \mid bc$, em que a, b e c são números inteiros positivos, então $a \mid b$ ou $a \mid c$.

9. Qual o quociente e o resto quando
- **a)** 19 é dividido por 7?
- **b)** -111 é dividido por 11?
- **c)** 789 é dividido por 23?
- **d)** 1001 é dividido por 13?
- **e)** 0 é dividido por 19?
- **f)** 3 é dividido por 5?
- **g)** -1 é dividido por 3?
- **h)** 4 é dividido por 1?

10. Qual o quociente e o resto quando
- **a)** 44 é dividido por 8?
- **b)** 777 é dividido por 21?
- **c)** -123 é dividido por 19?
- **d)** -1 é dividido por 23?
- **e)** -2002 é dividido por 87?
- **f)** 0 é dividido por 17?
- **g)** 1 234 567 é dividido por 1001?
- **h)** -100 é dividido por 101?

11. Considere m como um número inteiro positivo. Mostre que $a \equiv b \pmod{m}$ se $a \bmod m = b \bmod m$.

12. Considere m como um número inteiro positivo. Mostre que $a \bmod m = b \bmod m$ se $a \equiv b \pmod{m}$.

13. Mostre que se n e k são números inteiros positivos, então $\lceil n/k \rceil = \lfloor (n-1)/k \rfloor + 1$.

14. Mostre que se a é um número inteiro e d é um número inteiro positivo maior que 1, então o quociente e o resto obtidos quando a é dividido por d são $\lfloor a/d \rfloor$ e $a - d\lfloor a/d \rfloor$, respectivamente.

15. Encontre uma fórmula para o número inteiro com menor valor absoluto que é congruente ao número inteiro a módulo m, em que m é um número inteiro positivo.

16. Avalie as quantidades abaixo.
- **a)** $-17 \bmod 2$
- **b)** $144 \bmod 7$
- **c)** $-101 \bmod 13$
- **d)** $199 \bmod 19$

17. Avalie as quantidades abaixo.
- **a)** $13 \bmod 3$
- **b)** $-97 \bmod 11$
- **c)** $155 \bmod 19$
- **d)** $-221 \bmod 23$

18. Liste cinco números inteiros que são congruentes a 4 módulo 12.

19. Decida se cada um dos inteiros abaixo é congruente a 5 módulo 17.
- **a)** 80
- **b)** 103
- **c)** -29
- **d)** -122

20. Mostre que se $a \equiv b \pmod{m}$ e $c \equiv d \pmod{m}$, em que a, b, c, d e m são número inteiros com $m \geq 2$, então $a - c \equiv b - d \pmod{m}$.

21. Mostre que se $n \mid m$, em que n e m são números inteiros positivos maiores que 1 e se $a \equiv b \pmod{m}$, em que a e b são números inteiros, então $a \equiv b \pmod{n}$.

22. Mostre que se a, b, c e m são números inteiros, tal que $m \geq 2$, $c > 0$ e $a \equiv b \pmod{m}$, então $ac \equiv bc \pmod{mc}$.

23. Encontre contra-exemplos para cada uma das proposições abaixo sobre congruências.
- **a)** Se $ac \equiv bc \pmod{m}$, em que a, b, c e m são números inteiros com $m \geq 2$, então $a \equiv b \pmod{m}$.
- **b)** Se $a \equiv b \pmod{m}$ e $c \equiv d \pmod{m}$, em que a, b, c, d e m são números inteiros com c e d positivos e $m \geq 2$, então $a^c \equiv b^d \pmod{m}$.

24. Demonstre que se n é um número inteiro positivo e ímpar, então $n^2 \equiv 1 \pmod{8}$.

25. Mostre que se a, b, k e m são números inteiros, tal que $k \geq 1$, $m \geq 2$ e $a \equiv b \pmod{m}$, então $a^k \equiv b^k \pmod{m}$ sempre que k for um número inteiro positivo.

26. Quais localizações da memória são determinadas pela função de hashing $h(k) = k \bmod 101$ para os registros de seguro de uma companhia com os números do Seguro Social abaixo?
- **a)** 104578690
- **b)** 432222187
- **c)** 372201919
- **d)** 501338753

27. Um estacionamento tem 31 vagas para visitantes, numeradas de 0 a 30. Os visitantes são determinados a parar nas vagas usando-se a função de hashing $h(k) = k \bmod 31$, em que k é o número formado pelos três primeiros dígitos da placa do carro do visitante.
- **a)** Quais vagas são determinadas pela função de hashing para os carros que têm os seguintes três primeiros dígitos da placa do carro?

  317, 918, 007, 100, 111, 310
- **b)** Descreva um procedimento que os visitantes deverão seguir a fim de encontrar uma vaga livre para estacionar, quando o espaço designado a eles está ocupado.

28. Qual a seqüência de números pseudo-aleatórios gerada usando-se o gerador de congruência linear $x_{n+1} = (4x_n + 1) \bmod 7$ com origem $x_0 = 3$?

29. Qual a seqüência de números pseudo-aleatórios gerada usando-se o gerador multiplicativo puro $x_{n+1} = 3x_n \bmod 11$ com origem $x_0 = 2$?

30. Escreva um algoritmo em pseudocódigo para gerar uma seqüência de números pseudo-aleatórios usando o gerador de congruência linear.

31. Codifique a mensagem "DO NOT PASS GO" substituindo as letras por números, aplicando a função de codificação dada e, então, transcrevendo os números em letras.
- **a)** $f(p) = (p + 3) \bmod 26$ (o código de César)
- **b)** $f(p) = (p + 13) \bmod 26$
- **c)** $f(p) = (3p + 7) \bmod 26$

32. Decodifique as mensagens abaixo codificadas usando o código de César.
- **a)** EOXH MHDQV
- **b)** WHVW WRGDB
- **c)** HDW GLP VXP

Todos os livros são identificados por um **número de registro denominado ISBN,** um código com 13 dígitos $x_1x_2 \ldots x_{10}$,

determinado pela editora. Esses 13 dígitos consistem de blocos que identificam a linguagem, a editora, o número determinado para o livro pela editora e, por fim, um número com 1 dígito que é ou um dígito ou a letra X (usada para representar 10). Este último dígito é selecionado para que $\sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$ e é usado para detectar erros em dígitos individuais e transpor os dígitos.

33. Os primeiros nove dígitos de ISBN da versão européia da quinta edição deste livro são 0-07-119881. Qual é o último dígito para esse livro?

34. O ISBN da quinta edição de *Elementary Number Theory and Its Applications* é 0-32-123Q072, no qual Q é um dígito. Encontre o valor de Q.

35. Determine se o último dígito de ISBN para este livro foi computado corretamente pela editora.

3.5 Os Números Primos e os Máximos Divisores Comuns

Introdução

Na Seção 3.4, estudamos o conceito de divisibilidade de números inteiros. Um importante conceito baseado na divisibilidade é o de número primo. Um número primo é um número inteiro maior que 1 que é somente divisível por 1 e por ele mesmo. O estudo desses números remonta aos tempos antigos. Milhares de anos atrás, sabia-se que há infinitos números primos. Os maiores matemáticos dos últimos 400 anos estudaram-nos e demonstraram muitos resultados sobre eles. Entretanto, muitas conjecturas antigas sobre os números primos mantiveram-se sem resposta. Os números primos tornaram-se essenciais nos sistema de criptografia moderna.

Um teorema importante, o Teorema Fundamental da Aritmética, declara que todo número inteiro positivo pode ser escrito unicamente como o produto de números primos. A extensão de tempo necessária para a fatoração dos números inteiros grandes em seus fatores primos desempenha um importante papel na criptografia.

Nesta seção, estudaremos também os máximos divisores comuns de dois números inteiros, assim como o mínimo múltiplo comum de dois números inteiros. Vamos desenvolver um algoritmo para computar os máximos divisores comuns na Seção 3.6.

Números Primos

Todo número inteiro positivo maior que 1 é divisível por, no mínimo, dois números inteiros: por 1 e por ele mesmo. Os números inteiros positivos que têm exatamente dois fatores inteiros positivos diferentes são chamados de **primos.**

DEFINIÇÃO 1 Um número inteiro positivo p maior que 1 é chamado de *primo* se e somente se os fatores positivos de p forem 1 e p. Um número inteiro positivo que é maior que 1 e que não é primo é chamado de *composto.*

Lembre-se: O número inteiro n é composto se e somente se houver um número inteiro a, tal que $a \mid n$ e $1 < a < n$.

EXEMPLO 1 O número inteiro 7 é primo porque seus fatores positivos são 1 e 7, ao contrário do número inteiro 9, que é composto porque é divisível de 3. ◀

Os números primos menores que 100 são 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 e 97. Na Seção 7.6, estudaremos um procedimento, conhecido como **Crivo de Eratóstenes,** que pode ser usado para encontrar todos os primos não excedentes ao inteiro n.

Os números primos são os blocos de estruturas dos números inteiros positivos, como mostra o Teorema Fundamental da Aritmética. A demonstração será dada na Seção 4.2.

TEOREMA 1 **O TEOREMA FUNDAMENTAL DA ARITMÉTICA** Todo número inteiro positivo maior que 1 pode ser escrito unicamente como primo ou como o produto de dois ou mais primos, em que o valor dos fatores primos são escritos em ordem não decrescente de tamanho.

O Exemplo 2 dá algumas fatorações em números primos de números inteiros.

EXEMPLO 2 As fatorações em números primos de 100, 641, 999 e 1024 são dadas por

$$100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 5^2,$$
$$641 = 641,$$
$$999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37,$$
$$1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}.$$ ◀

É importante mostrar que um dado número inteiro é primo. Por exemplo, em criptologia, um grande número de primos é usado em alguns métodos para construir mensagens secretas. Um procedimento para mostrar que um número inteiro é primo é baseado na seguinte observação.

TEOREMA 2 Se n é um número inteiro composto, então n tem um divisor primo menor ou igual a $\sqrt{n}$.

Demonstração: Se n é composto, pela definição de um número inteiro composto, sabemos que n tem um fator a com $1 < a < n$. Assim, pela definição de fator de um número inteiro positivo, temos que $n = ab$, em que b é um número inteiro positivo maior que 1. Mostraremos que $a \leq \sqrt{n}$ ou $b \leq \sqrt{n}$. Se $a > \sqrt{n}$ e $b > \sqrt{n}$, então $ab > \sqrt{n} \cdot \sqrt{n} = n$, que é uma contradição. Conseqüentemente, $a \leq \sqrt{n}$ ou $b \leq \sqrt{n}$. Como ambos, a e b, são divisores de n, vemos que n tem um divisor positivo não excedente a $\sqrt{n}$. Este divisor é primo ou, pelo Teorema Fundamental da Aritmética, tem um divisor primo menor que ele mesmo. Neste caso, n tem um divisor primo menor ou igual a $\sqrt{n}$. ◁

A partir do Teorema 2, temos que um número inteiro é primo se ele não for divisível por qualquer primo menor que ou igual a sua raiz quadrada. No exemplo a seguir, esta observação é usada para mostrar que 101 é primo.

EXEMPLO 3 Mostre que 101 é primo.

Solução: Os únicos primos não excedentes a $\sqrt{101}$ são 2, 3, 5 e 7. Como 101 não é divisível por 2, 3, 5 ou 7 (o quociente de 101 e cada um desses números inteiros não é um inteiro), temos que 101 é primo. ◀

Como cada número inteiro tem uma fatoração em números primos, seria útil ter um procedimento para encontrar essa fatoração. Considere o problema de encontrar a fatoração de n. Comece dividindo n por sucessivos números primos, começando com o menor, 2. Se n tiver um fator primo, então, pelo Teorema 3, será encontrado um fator p não excedente a $\sqrt{n}$. Logo, se não encontrarmos um fator primo não excedente a $\sqrt{n}$, então n é primo. Por outro lado, se um fator primo p for encontrado, continue com a fatoração n/p. Note que n/p não tem fatores primos menores que p. Novamente, se n/p não tem fator primo maior que ou igual a p não excedente à sua raiz quadrada, então ele é primo. Por outro lado, se ele tiver um fator primo q, continue com a fatoração $n/(pq)$. Este procedimento, que é mostrado no Exercício 4, continua até que a fatoração seja reduzida a um primo.

EXEMPLO 4 Encontre a fatoração em números primos de 7007.

Solução: Para fazer isso, primeiro realize divisões de 7007 por primos sucessivos, começando com o 2. Nenhum dos primos 2, 3 e 5 divide 7007. Entretanto, 7 divide 7007, com $7007/7 = 1001$. Depois, divida 1001 por sucessivos primos, começando com 7. Vemos imediatamente que 7 também divide 1001, pois $1001/7 = 143$. Continue dividindo 143 por sucessivos primos, começando com 7. Embora 7 não divida 143, 11 divide 143, e $143/11 = 13$. Como 13 é um número primo, o procedimento está completo. Temos que a fatoração em números primos de 7007 é $7 \cdot 7 \cdot 11 \cdot 13 = 7^2 \cdot 11 \cdot 13$. ◀

Os números primos foram estudados em tempos remotos por razões filosóficas. Hoje, há muitas razões práticas para estudá-los. Em particular, um grande número de primos desempenha um papel crucial na criptologia, como veremos na Seção 3.7.

PRIMOS INFINITOS Sabe-se que há um número infinito de primos. Isto significa que sempre que $p_1, p_2, \ldots, p_n$ forem os n menores números primos, sabemos que há um primo maior não listado. Apresentaremos este fato usando uma demonstração dada por Euclides em seu famoso texto, *Os Elementos*.

TEOREMA 3 Há um número infinito de primos.

Demonstração: Apresentaremos este teorema usando uma demonstração por contradição. Suponhamos que haja apenas um número finito de primos, $p_1, p_2, \ldots, p_n$. Considere

$$Q = p_1 p_2 \cdots p_n + 1.$$

Pelo Teorema Fundamental da Aritmética, Q não é primo se puder ser escrito como o produto de dois ou mais primos. Entretanto, nenhum dos números primos p_j divide Q, pois se $p_j \mid Q$, então p_j divide $Q - p_1 p_2 \cdots p_n = 1$. Assim, há um número primo fora da lista $p_1, p_2, \ldots, p_n$. Este primo é ou Q, se ele for um primo, ou um fator primo de Q. Isto é uma contradição, porque supusemos que tínhamos listado todos os números primos. Conseqüentemente, há um número infinito de primos. (Note que nesta demonstração *não* afirmamos que Q é primo! Além disso, nesta demonstração, demos uma demonstração de existência não construtiva de que dada qualquer lista de números primos, há um primo que não está na lista. Para esta demonstração ser construtiva, poderíamos ter tomado um primo explicitamente não em nossa lista original de n números primos.) ◁

Como há um número infinito de primos, dado qualquer número inteiro positivo, há números primos maiores que este número inteiro. Há uma busca constante para descobrir números primos cada vez maiores; por quase todos esses 300 anos, o maior primo conhecido foi um número inteiro de formato especial $2^p - 1$, em que p é também primo. Tais primos são chamados de **primos de Mersenne,** em homenagem ao monge francês Marin Mersenne, que os estudou no século XVII. A razão pela qual o maior primo conhecido tem sido geralmente um primo de Mersenne é que há um teste extremamente eficiente, conhecido como teste de Lucas–Lehmer, para determinar se $2^p - 1$ é um número primo. Além disso, nem sempre é possível testar números deste ou de outros certos formatos tão rapidamente quanto determinar se eles são primos.

EXEMPLO 5 Os números $2^2 - 1 = 3$, $2^3 - 1 = 7$ e $2^5 - 1 = 31$ são primos de Mersenne, enquanto $2^{11} - 1 = 2047$ não é um primo de Mersenne, pois $2047 = 23 \cdot 89$. ◀

O progresso em encontrar os primos de Mersenne tem sido constante desde que os computadores foram inventados. No começo de 2006, 43 primos de Mersenne eram conhecidos, em comparação a 1990, que eram apenas 12. O maior primo de Mersenne conhecido (novamente, até 2006) é $2^{30\,402\,457} - 1$, um número com mais de nove milhões de dígitos, que foi mostrado como número primo em dezembro de 2005. Um grande esforço é feito pelo Great Internet Mersenne Prime Search (GIMPS) na pesquisa de novos primos de Mersenne, inclusive com implicações práticas. Um teste de controle de qualidade para super-computadores foi fazer uma cópia do teste de Lucas–Lehmer que estabelece um novo grande primo de Mersenne.

A DISTRIBUIÇÃO DOS NÚMEROS PRIMOS O Teorema 3 nos diz que há um número infinito de primos. Entretanto, quantos números primos são menores que o número positivo x? Esta questão tem interessado aos matemáticos há muitos anos; no século XVIII, eles produziram tabelas enormes de números primos para reunir evidências sobre sua distribuição. Usando essa evidência, os grandes matemáticos da época, incluindo Gauss e Legendre, conjecturaram, mas não demonstraram, o Teorema 4.

TEOREMA 4

O TEOREMA DO NÚMERO PRIMO A razão do número de primos não excedente a x e $x/\ln x$ tende a 1 quando x cresce sem limitante. (Aqui, $\ln x$ é o logaritmo natural de x.)

Links

O Teorema do Número Primo foi demonstrado pela primeira vez em 1896 pelo matemático francês Jacques Hadamard e pelo matemático belga Charles-Jean-Gustave-Nicholas de la Valleé-Poussin, usando a teoria das variáveis complexas. Embora demonstrações sem o uso dessas variáveis tenham sido descobertas, todas as demonstrações conhecidas do Teorema do Número Primo são bem complicadas.

Podemos usar o Teorema do Número Primo para estimar a probabilidade de um número escolhido aleatoriamente ser primo. O Teorema do Número Primo nos diz que o número de primos não excedentes a x *é* aproximadamente $x/\ln x$. Conseqüentemente, a probabilidadede um número inteiro positivo escolhido aleatoriamente menor que n ser primo é aproximadamente $(n/\ln n)/n = 1/\ln n$. Às vezes precisamos encontrar um número primo com determinado número de dígitos. Gostaríamos de uma estimativa de quantos números inteiros com determinado número de dígitos precisaremos selecionar antes de encontrar um primo. Usando o Teorema do Número Primo e cálculo, podemos mostrar que a probabilidade de um número inteiro n ser primo é, também, aproximadamente $1/\ln n$. Por exemplo, a probabilidade de um número inteiro próximo de 10^{1000} ser primo é aproximadamente $1/\ln 10^{1000}$, que é aproximadamente 1/2300. (Claro que, escolhendo apenas números ímpares, dobramos nossas chances de encontrar um primo.)

Links

MARIN MERSENNE (1588–1648) Mersenne nasceu em Maine, na França, em uma família de trabalhadores e freqüentou a Faculdade de Mans e a Faculdade Jesuíta em La Flèche. Ele continuou sua educação em Sorbonne, estudando teologia de 1609 a 1611. Entrou para a ordem religiosa dos frades Minims em 1611, um grupo cujo nome vem da palavra *minimi* (os membros deste grupo eram extremamente humildes; eles consideravam-se a menor de todas as ordens religiosas). Além de rezar, os membros deste grupo devotavam sua energia ao estudo. Em 1612, ele se tornou um padre no Palácio Real em Paris; entre 1614 e 1618, ensinou filosofia no Convento Minims em Nevers. Retornou a Paris em 1619, onde sua cela na l'Annociade dos Minims tornou-se um local de encontro de cientistas, filósofos e matemáticos franceses, incluindo Fermat e Pascal. Mersenne correspondeu-se intensamente com os acadêmicos de toda a Europa, servindo como intermediário para o conhecimento matemático e científico, uma função que depois foi desempenhada pelos periódicos matemáticos (e hoje também pela Internet). Mersenne escreveu livros sobre mecânica, física matemática, matemática, música e acústica. Ele estudou os números primos e tentou, sem sucesso, construir uma fórmula para representar todos eles. Em 1644, Mersenne declarou que $2^p - 1$ é primo para $p = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257$, mas é composto para todos os outros primos menores que 257. Demorou mais de 300 anos para determinar que a afirmação de Mersenne estava errada cinco vezes. Especificamente, $2^p - 1$ não é primo para $p = 67$ e $p = 257$, mas é primo para $p = 61, p = 87$ e $p = 107$. É também notável que Mersenne defendeu das críticas religiosas dois dos homens mais famosos de seu tempo, Descartes e Galileu. Ele também ajudou a taxar alquimistas e astrólogos como fraudes.

Usar a tentativa de divisão com o Teorema 2 nos dá procedimentos para fatoração e testes a fim de verificar se um número é primo. Entretanto, esses procedimentos não são algoritmos eficientes; algoritmos muito mais práticos e eficientes que esses foram desenvolvidos. A fatoração e o teste para verificar se um número é primo tornaram-se importantes nas aplicações da teoria dos números para a criptografia. Isto levou a um grande interesse em desenvolver algoritmos eficientes para as duas tarefas. Procedimentos mais inteligentes foram desenvolvidos nos últimos 30 anos para gerar números primos grandes de forma mais eficiente. Entretanto, mesmo os métodos de fatoração mais poderosos tendo sido desenvolvidos ao mesmo tempo, fatorar grandes números demora um tempo extraordinário. No entanto, o desafio de fatorar grandes números interessa a muitas pessoas. Há um grande esforço na Internet para fatorar números grandes, especialmente aqueles com formato especial de $k^n \pm 1$, em que k é um número inteiro positivo pequeno e n é um número inteiro positivo grande (tais números são chamados de *números de Cunningham*). De tempos em tempos, há uma lista dos "Dez mais Procurados" para a fatoração de números muito grandes desse tipo.

Conjecturas e Problemas Abertos sobre Números Primos

A teoria dos números é vista como um assunto fácil de formular conjecturas, algumas das quais são difíceis de ser demonstradas e outras que permaneceram como problemas abertos por muitos anos. Vamos descrever algumas conjecturas na teoria dos números e discutiremos seus status nos exemplos de 6 a 9.

EXEMPLO 6 Seria útil ter uma função $f(n)$ tal que $f(n)$ fosse um número primo para todos os números inteiros positivos n. Se tivéssemos tal função, poderíamos encontrar os maiores números primos para usar em criptografia e em outras aplicações. Procurar por tal função nos faz cair em diferentes funções polinomiais, assim como fizeram alguns matemáticos centenas de anos atrás. Depois de muita informação de computação, podemos encontrar o polinômio $f(n) = n^2 - n + 41$. Este polinômio tem uma propriedade interessante de que $f(n)$ é primo para todos os números inteiros positivos n não excedentes a 40. [Temos $f(1) = 41, f(2) = 43, f(3) = 47, f(4) = 53$ e assim por diante.] Isto pode nos levar a uma conjectura de que $f(n)$ é primo para todos os n números inteiros positivos. Podemos sustentar esta conjectura?

Exemplos Extras

Solução: Talvez não cause surpresa que tal conjectura tende a ser falsa; não temos que ir longe para encontrar um número inteiro positivo n para o qual $f(n)$ é composto, pois $f(41) = 41^2 - 41 + 41 = 41^2$. Como $f(n) = n^2 - n + 41$ é um número primo para todos os n números inteiros positivos, com $1 \leq n \leq 40$, poderíamos estar tentados a encontrar um polinômio diferente com a propriedade de que $f(n)$ é um número primo para *todos* os n números inteiros positivos. Entretanto, não há tal polinômio. Podemos mostrar que para todo polinômio $f(n)$ com coeficientes inteiros, há um número inteiro positivo y tal que $f(y)$ é composto. (Veja o Exercício 37 dos Exercícios Complementares no final deste capítulo.) ◀

Muitos problemas famosos sobre números primos ainda esperam por resoluções de pessoas brilhantes. Descrevemos uma pequena parte desses problemas abertos mais acessíveis e mais bem conhecidos nos exemplos 7 a 9. A teoria dos números é conhecida por sua facilidade de construir conjecturas de fácil entendimento que resistem aos ataques de todos, das técnicas mais sofisticadas ou, simplesmente, resistem a todos os ataques. Apresentamos essas conjecturas para mostrar que muitas questões que parecem relativamente simples permanecem sem resolução mesmo no século XXI.

EXEMPLO 7 **Conjectura de Goldbach** Em 1742, Christian Goldbach, em uma carta a Leonhard Euler, conjecturou que todo número inteiro ímpar n, $n > 5$, é a soma de três primos. Euler respondeu que esta conjectura equivale a dizer que todo número inteiro par n, $n > 2$, é a soma de dois primos (veja o Exercício 39 dos Exercícios Complementares no final deste capítulo). A conjectura de que todo número inteiro par n, $n > 2$, é a soma de dois primos é conhecida como **conjectura de Goldbach.** Podemos checar esta conjectura com números pares pequenos. Por exemplo, $4 = 2 + 2$, $6 = 3 + 3$, $8 = 5 + 3$, $10 = 7 + 3$, $12 = 7 + 5$ e assim por diante. A conjectura de Goldbach era verificada com cálculos manuais para números relativamente pequenos, mas passou para milhões com o advento do computador. Com um computador, podemos checar para números extremamente grandes. Até o início de 2006, a conjectura havia sido checada para todos os números inteiros positivos pares até $2 \cdot 10^{17}$.

Links

Embora não tenha sido encontrada uma demonstração da conjectura de Goldbach, muitos matemáticos acreditam que ela seja verdadeira. Vários teoremas foram demonstrados usando métodos complicados de análise da teoria dos números que estão fora do alcance deste livro, estabelecendo resultados mais fracos que a conjectura de Goldbach. Entre esses, há o resultado de que todo número inteiro positivo maior que 2 é a soma de, no máximo, seis números primos (demonstrado em 1995 por O. Ramaré) e que todo número inteiro positivo suficientemente grande é a soma de um primo e de um número que ou é primo ou é o produto de dois primos (demonstrado em 1966 por J. R. Chen). Talvez a conjectura de Goldbach seja comprovada num futuro não muito distante. ◀

EXEMPLO 8 Há muitas conjecturas que afirmam que há um número infinito de primos de determinadas formas especiais. Uma conjectura deste tipo é aquela que diz que existem infinitos primos na forma $n^2 + 1$, em que n é um número inteiro positivo. Por exemplo, $5 = 2^2 + 1$, $17 = 4^2 + 1$, $37 = 6^2 + 1$ e assim por diante. O melhor resultado conhecido é que há um número infinito de números inteiros positivos n tal que $n^2 + 1$ é primo ou é o produto de, no máximo, dois primos (demonstrado por Henryk Iwaniec em 1973 usando técnicas avançadas de teoria analítica dos números, que está fora do alcance deste livro). ◀

EXEMPLO 9 **Conjectura dos Primos Gêmeos** **Primos Gêmeos** são números que diferenciam-se por duas unidades, tal como 3 e 5, 5 e 7, 11 e 13, 17 e 19, 4967 e 4969. A conjectura dos primos gêmeos afirma que há um número infinito deles. O maior resultado demonstrado sobre os primos gêmeos é que há um número infinito de pares p e $p + 2$, em que p é um número primo e $p + 2$ é também primo ou o produto de dois primos (demonstrado por J. R. Chen em 1966). O recorde mundial para primos gêmeos, até o início de 2006, consistia nos números $16\,869\,987\,339\,975 \cdot 2^{171\,960} \pm 1$, números com 51 779 dígitos. ◀

Os Máximos Divisores Comuns e os Mínimos Múltiplos Comuns

O maior número inteiro que divide dois números inteiros é chamado de **máximo divisor comum** desses inteiros.

DEFINIÇÃO 2 Considere a e b como números inteiros diferentes de zero. O maior número inteiro d, tal que $d \mid a$ e $d \mid b$ é chamado de *máximo divisor comum* de a e b. O máximo divisor comum de a e b é indicado por $\text{mdc}(a, b)$.

O máximo divisor comum de dois números inteiros, diferentes de zero, existe porque o conjunto dos divisores comuns desses inteiros é finito. Uma maneira de encontrar o máximo divisor comum de dois números inteiros é encontrar os divisores positivos comuns a ambos os inteiros e então pegar o maior divisor. Isto é feito nos exemplos 10 e 11. Mais a frente, um método mais eficiente para encontrar máximos divisores comuns será apresentado.

EXEMPLO 10 Qual é o máximo divisor comum de 24 e 36?

Solução: Os divisores positivos comuns de 24 e 36 são 1, 2, 3, 4, 6 e 12. Assim, $\text{mdc}(24, 36) = 12$. ◀

CHRISTIAN GOLDBACH (1690–1764) Christian Goldbach nasceu em Königsberg, Prússia, na cidade conhecida pelo seu famoso problema das pontes (que será estudado na Seção 9.5). Ele tornou-se professor de matemática na Academia de São Petersburgo, em 1725. Em 1728, Goldbach foi para Moscou para ser tutor do filho do Czar. Entrou para o mundo da política e, em 1742, tornou-se membro do Ministério de Relações Internacionais russo. Goldbach é conhecido por sua correspondência com matemáticos famosos, incluindo Euler e Bernoulli, por sua conjectura em teoria dos números e por diversas contribuições em análise.

EXEMPLO 11 Qual é o máximo divisor comum de 17 e 22?

Solução: Os números inteiros 17 e 22 não têm divisores positivos comuns a não ser 1, então $\text{mdc}(17, 22) = 1$. ◀

Como freqüentemente é importante salientar quando dois números inteiros não têm divisor positivo comum além do 1, temos a Definição 3.

DEFINIÇÃO 3 Os números inteiros a e b são *relativamente primos,* ou *primos entre si,* se o máximo divisor comum deles for 1.

EXEMPLO 12 Pelo Exemplo 11, temos que os números inteiros 17 e 22 são relativamente primos, pois $\text{mdc}(17, 22) = 1$. ◀

Como freqüentemente precisamos especificar se cada par de números inteiros em um conjunto de inteiros não tem um divisor positivo comum além de 1, temos a Definição 4.

DEFINIÇÃO 4 Os números inteiros $a_1, a_2, \ldots, a_n$ são *pares relativamente primos,* ou *primos entre si dois a dois,* se $\text{mdc}(a_i, a_j) = 1$ sempre que $1 \leq i < j \leq n$.

EXEMPLO 13 Determine se os números inteiros 10, 17 e 21 são pares relativamente primos. Faça o mesmo com os inteiros 10, 19 e 24.

Solução: Como $\text{mdc}(10, 17) = 1$, $\text{mdc}(10, 21) = 1$ e $\text{mdc}(17, 21) = 1$, concluímos que 10, 17 e 21 são pares relativamente primos.

Como $\text{mdc}(10, 24) = 2 > 1$, vemos que 10, 19 e 24 não são pares relativamente primos. ◀

Outra maneira de encontrar o máximo divisor comum de dois números inteiros é usar a fatoração desses inteiros. Suponha que as fatorações em números primos dos inteiros a e b, diferente de zero, são

$$a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}, \; b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n},$$

em que cada expoente é um número inteiro não negativo e todos os primos que aparecem na fatoração em números primos de a ou b estão inclusos nas duas fatorações, com expoente zero se necessário. Então, $\text{mdc}(a, b)$ é dado por

$$\text{mdc}(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \cdots p_n^{\min(a_n, b_n)},$$

em que $\min(x, y)$ representa o menor entre os números x e y. Para declarar que esta fórmula para $\text{mdc}(a, b)$ é válida, devemos mostrar que o número inteiro do lado direito divide a e b e que nenhum inteiro maior divide. Este inteiro divide a e b porque a potência de cada primo na fatoração não excede a potência deste primo na fatoração de a nem de b. Além disso, nenhum número inteiro maior pode dividir a e b, porque os expoentes dos números primos nesta fatoração não podem ser aumentados e nenhum primo pode ser incluído.

EXEMPLO 14 Como a fatoração em números primos de 120 e 500 é $120 = 2^3 \cdot 3 \cdot 5$ e $500 = 2^2 \cdot 5^3$, o máximo divisor comum é

$$\text{mdc}(120, 500) = 2^{\min(3, 2)} 3^{\min(1, 0)} 5^{\min(1, 3)} = 2^2 3^0 5^1 = 20.$$ ◀

A fatoração em números primos também pode ser usada para encontrar o **mínimo múltiplo comum** de dois números inteiros.

DEFINIÇÃO 5 O *mínimo múltiplo comum* dos números inteiros positivos a e b é o menor inteiro positivo divisível por ambos, a e b. O mínimo múltiplo comum de a e b é indicado por $\text{mmc}(a, b)$.

O mínimo múltiplo comum existe porque o conjunto dos números inteiros divisíveis por a e b não é vazio e todo conjunto de números inteiros positivos não vazio tem um menor elemento (pela propriedade da boa ordenação, que será estudada na Seção 4.2). Suponha que a fatoração em números primos de a e b seja como antes. Então, o mínimo múltiplo comum de a e b é dado por

$$\text{mmc}(a, b) = p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \cdots p_n^{\max(a_n, b_n)}$$

em que $\max(x, y)$ indica o máximo de dois números x e y. Esta fórmula é válida porque um múltiplo comum de a e b tem, pelos menos, máx.(a_i, b_i) fatores de p_i em sua fatoração em números primos, e o mínimo múltiplo comum não tem outros fatores primos além daquele em a e b.

EXEMPLO 15 Qual é o mínimo múltiplo comum de $2^3 3^5 7^2$ e $2^4 3^3$?

Solução: Temos

$$\text{mmc}(2^3 3^5 7^2, 2^4 3^3) = 2^{\max(3,4)} 3^{\max(5,3)} 7^{\max(2,0)} = 2^4 3^5 7^2. \quad \blacktriangleleft$$

O Teorema 5 dá a relação entre o máximo divisor comum e o mínimo múltiplo comum de dois números inteiros. Ele pode ser demonstrado usando-se a fórmula empregada para essas quantidades. A demonstração desse teorema foi deixada como um exercício para o leitor.

TEOREMA 5 Considere a e b como números inteiros positivos. Então,

$$ab = \text{mdc}(a, b) \cdot \text{mmc}(a, b).$$

Exercícios

1. Determine se cada um destes números inteiros é primo.
a) 21 **b)** 29 **c)** 71
d) 97 **e)** 111 **f)** 143

2. Determine se cada um destes números inteiros é primo.
a) 19 **b)** 27 **c)** 93
d) 101 **e)** 107 **f)** 113

3. Encontre a fatoração em números primos de cada um destes números inteiros.
a) 88 **b)** 126 **c)** 729
d) 1001 **e)** 1111 **f)** 909 090

4. Encontre a fatoração em números primos de cada um destes números inteiros.
a) 39 **b)** 81 **c)** 101
d) 143 **e)** 289 **f)** 899

5. Encontre a fatoração de números primos de 10!.

***6.** Quantos zeros há no final de 100!?

***7.** Mostre que $\log_2 3$ é um número irracional. Lembre-se de que um número irracional é um número real x que não pode ser escrito como a razão de dois números inteiros.

8. Demonstre que para todo número inteiro positivo n existem n números inteiros compostos consecutivos. [*Dica:* Considere os n números inteiros consecutivos começando com $(n + 1)! + 2$.]

***9.** Demonstre ou negue a existência de três números inteiros positivos e ímpares consecutivos que são primos, ou seja, números ímpares primos na forma p, $p + 2$ e $p + 4$.

10. Quais números inteiros positivos menores que 12 são relativamente primos de 12?

11. Quais números inteiros positivos menores que 30 são relativamente primos de 30?

12. Determine se os números inteiros em cada um dos conjuntos abaixo são pares relativamente primos.
a) 21, 34, 55 **b)** 14, 17, 85
c) 25, 41, 49, 64 **d)** 17, 18, 19, 23

13. Determine se os números inteiros em cada um dos conjuntos abaixo são pares relativamente primos.
a) 11, 15, 19 **b)** 14, 15, 21
c) 12, 17, 31, 37 **d)** 7, 8, 9, 11

14. Chamamos um número inteiro positivo de **perfeito** se ele for igual à soma de seus divisores positivos diferentes dele mesmo.
a) Mostre que 6 e 28 são perfeitos.
b) Mostre que $2^{p-1}(2^p - 1)$ é um número perfeito quando $2^p - 1$ for primo.

15. Mostre que se $2^n - 1$ for primo, então n é primo. [*Dica:* Use a identidade $2^{ab} - 1 = (2^a - 1) \cdot (2^{a(b-1)} + 2^{a(b-2)} + \cdots + 2^a + 1)$.]

16. Determine se cada um dos números inteiros abaixo é primo, verificando algumas das declarações de Mersenne.
a) $2^7 - 1$ **b)** $2^9 - 1$
c) $2^{11} - 1$ **d)** $2^{13} - 1$

O valor da **função ϕ de Euler** para o número inteiro positivo n é definido como sendo o número de inteiros positivos menores que ou iguais a n que são relativamente primos de n. (*Nota:* ϕ é a letra grega "fi".)

17. Encontre
a) $\phi(4)$. **b)** $\phi(10)$. **c)** $\phi(13)$.

18. Mostre que n é primo se e somente se $\phi(n) = n - 1$.

19. Qual é o valor de $\phi(p^k)$ quando p é primo e k é um número inteiro positivo?

20. Quais são os máximos divisores comuns de cada par de números inteiros abaixo?
a) $2^2 \cdot 3^3 \cdot 5^5, 2^5 \cdot 3^3 \cdot 5^2$
b) $2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13, 2^{11} \cdot 3^9 \cdot 11 \cdot 17^{14}$
c) $17, 17^{17}$ **d)** $2^2 \cdot 7, 5^3 \cdot 13$
e) $0, 5$ **f)** $2 \cdot 3 \cdot 5 \cdot 7, 2 \cdot 3 \cdot 5 \cdot 7$

21. Quais são os máximos divisores comuns de cada par de números inteiros abaixo?
a) $3^7 \cdot 5^3 \cdot 7^3, 2^{11} \cdot 3^5 \cdot 5^9$ **b)** $11 \cdot 13 \cdot 17, 2^9 \cdot 3^7 \cdot 5^5 \cdot 7^3$
c) $23^{31}, 23^{17}$ **d)** $41 \cdot 43 \cdot 53, 41 \cdot 43 \cdot 53$
e) $3^{13} \cdot 5^{17}, 2^{12} \cdot 7^{21}$ **f)** $1111, 0$

22. Qual é o mínimo múltiplo comum de cada par do Exercício 20?

23. Qual é o mínimo múltiplo comum de cada par do Exercício 21?

24. Encontre mdc(1000, 625) e mmc(1000, 625) e verifique se mdc(1000, 625) · mmc(1000, 625) = 1000 · 625.

25. Encontre mdc(92928, 123552) e mmc(92928, 123552) e verifique se mdc(92928, 123552) · mmc(92928, 123552) = 92928 · 123552. [*Dica:* Primeiro encontre as fatorações em números primos de 92928 e 123552.]

26. Se o produto de dois número inteiros é $2^7 3^8 5^2 7^{11}$ e seu máximo divisor comum é $2^3 3^4 5$, qual é o mínimo múltiplo comum entre eles?

27. Mostre que se a e b forem números inteiros positivos, então ab = mdc(a,b) · mmc(a b). [*Dica:* Use as fatorações em números primos de a e b e as fórmulas para mdc(a, b) e mmc(a, b) em termos dessas fatorações.]

28. Encontre o menor número inteiro positivo com n fatores diferentes quando n for
a) 3. **b)** 4.
c) 5. **d)** 6.
e) 10.

29. Você pode encontrar uma fórmula ou regra para o n-ésimo termo de uma seqüência relativa aos números primos ou à fatoração em números primos para que os termos iniciais da seqüência tenham os valores abaixo?
a) 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, ...
b) 1, 2, 3, 2, 5, 2, 7, 2, 3, 2, 11, 2, 13, 2, ...
c) 1, 2, 2, 3, 2, 4, 2, 4, 3, 4, 2, 6, 2, 4, ...
d) 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, ...
e) 1, 2, 3, 3, 5, 5, 7, 7, 7, 11, 11, 13, 13, ...
f) 1, 2, 6, 30, 210, 2310, 30030, 510510, 9699690, 223092870, ...

30. Você pode encontrar uma fórmula ou regra para o n-ésimo termo de uma seqüência relativa aos números primos ou à fatoração em números primos para que os termos iniciais da seqüência tenham os valores abaixo?
a) 2, 2, 3, 5, 5, 7, 7, 11, 11, 11, 11, 13, 13, ...
b) 0, 1, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 6, 6, ...
c) 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, ...
d) 1, −1, −1, 0, −1, 1, −1, 0, 0, 1, −1, 0, −1, 1, 1, ...
e) 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, ...
f) 4, 9, 25, 49, 121, 169, 289, 361, 529, 841, 961, 1369, ...

31. Demonstre que o produto de três números inteiros consecutivos quaisquer é divisível por 6.

32. Mostre que se a, b e m são números inteiros tal que $m \geq 2$ e $a \equiv b \pmod{m}$, então mdc(a, m) = mdc(b, m).

***33.** Demonstre ou negue que $n^2 - 79n + 1601$ é primo sempre que n for um número inteiro positivo.

34. Demonstre ou negue que $p_1 p_2 \cdots p_n + 1$ é primo para todo número inteiro positivo n, em que $p_1, p_2, \ldots, p_n$ são os n menores números primos.

35. Adapte a demonstração no texto da existência de um número infinito de primos para uma demonstração na qual existe um número infinito de primos na forma $4k + 3$, em que k é um número inteiro não negativo. [*Dica:* Suponha que há apenas tais primos $q_1, q_2, \ldots, q_n$, e considere o número $4q_1 q_2 \cdots q_n - 1$.]

***36.** Demonstre que o conjunto dos números racionais positivos é contável montando uma função que determine para um número racional p/q com mdc(p, q) = 1 a base 11 a partir da representação decimal de p seguido do dígito A de base 11, que corresponde ao número decimal 10, seguido de uma representação decimal de q.

***37.** Demonstre que o conjunto dos números racionais positivos é contável mostrando que a função K é injetora entre o conjunto dos números racionais positivos e o conjunto dos números inteiros positivos, se $K(m/n) = p_1^{2a_1} p_2^{2a_2} \cdots p_s^{2a_s} q_1^{2b_1 - 1} q_2^{2b_2 - 1} \cdots q_t^{2b_t - 1}$, em que mdc($m$, n) = 1 e as fatorações em números primos de m e n são $m = p_1^{a_1} p_2^{a_2} \cdots p_s^{a_s}$ e $n = q_1^{b_1} q_2^{b_2} \cdots q_t^{b_t}$.

3.6 Os Números Inteiros e os Algoritmos

Introdução

Como foi mencionado na Seção 3.1, o termo *algoritmo* originariamente referia-se aos procedimentos de realização de operações aritméticas usando-se a representação decimal dos números inteiros. Esses algoritmos, adaptados ao uso com representações binárias, são a base da aritmética computacional. Eles fornecem boas ilustrações do conceito de algoritmo e de sua complexidade. Por essas razões, eles serão discutidos nesta seção.

Há muitos algoritmos importantes que envolvem números inteiros além daqueles usados em aritmética, incluindo o algoritmo de Euclides, que é um dos mais usados e talvez o mais antigo em matemática. Descreveremos também um algoritmo que permite encontrar a expansão na base b de um número inteiro positivo para qualquer base b e para a potenciação modular, um algoritmo importante em criptologia.

Representações de Números Inteiros

No cotidiano usamos a notação decimal para representar os números inteiros. Por exemplo, 965 é usado para indicar $9 \cdot 10^2 + 6 \cdot 10 + 5$. Entretanto, é conveniente usar bases diferentes de 10. Em particular, os computadores geralmente usam a notação binária (com base 2) quando trabalham com aritmética, e notação octal (base 8) ou hexadecimal (base 16) quando representam caracteres, tais como letras ou dígitos. No entanto, podemos usar qualquer número inteiro positivo maior que 1 como base quando indicamos números inteiros. Isto é estabelecido pelo Teorema 1.

TEOREMA 1 Considere b como um número inteiro positivo maior que 1. Então, se n for um número inteiro positivo, pode ser expresso unicamente na forma

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0,$$

em que k é um inteiro não negativo, $a_0, a_1, \ldots, a_k$ são números inteiros não negativos menores que b e $a_k \neq 0$.

Uma demonstração desse teorema pode ser feita por meio da indução matemática, um método de demonstração que será discutido na Seção 4.1 e que pode ser encontrado também em [Ro05]. A representação de n dada no Teorema 1 é chamada de **expansão na base b de n.** A expansão na base b de n é indicada por $(a_k a_{k-1} \ldots a_1 a_0)_b$. Por exemplo, $(245)_8$ representa $2 \cdot 8^2 + 4 \cdot 8 + 5 = 165$.

EXPANSÕES BINÁRIAS Ao escolher 2 como base, temos as **expansões binárias** dos números inteiros. Na notação binária, cada dígito é 0 ou 1. Em outras palavras, a expansão binária de um número inteiro é apenas uma cadeia de bits. As expansões binárias (e expansões que são variantes das expansões binárias) são usadas por computadores para representar e executar aritmética com números inteiros.

EXEMPLO 1 Qual é a expansão decimal do número inteiro que tem $(1\ 0101\ 1111)_2$ como sua expansão binária?

Solução: Temos

$$(1\ 0101\ 1111)_2 = 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 351. \quad \blacktriangleleft$$

EXPANSÕES HEXADECIMAIS Dezesseis é outra base usada em ciência da computação. A expansão na base 16 de um número inteiro é chamada de expansão **hexadecimal**. Dezesseis dígitos diferentes são usados para representar tais expansões. Geralmente, os dígitos usados são 0,

1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F, em que as letras de A a F representam os dígitos correspondentes aos números de 10 a 15 (em notação decimal).

EXEMPLO 2 Qual é a expansão decimal da expansão hexadecimal de $(2AE0B)_{16}$?

Solução: Temos

$$(2AE0B)_{16} = 2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16 + 11 = (175627)_{10}.$$ ◀

Cada dígito hexadecimal pode ser representado usando-se quatro bits. Por exemplo, vemos que $(1110 \quad 0101)_2 = (E5)_{16}$, pois $(1110)_2 = (E)_{16}$ e $(0101)_2 = (5)_{16}$. **Bytes**, que são cadeias de bits de extensão oito, podem ser representados por dois dígitos hexadecimais.

CONVERSÃO DE BASE Vamos descrever agora um algoritmo para a construção da expansão na base b de um número inteiro n. Primeiro, dividimos n (escrito na base 10) por b para obter um quociente e um resto, ou seja,

$$n = bq_0 + a_0, \qquad 0 \leq a_0 < b.$$

O resto, a_0, é o dígito mais à direita da expansão na base b de n. Depois, dividimos q_0 por b para obter

$$q_0 = bq_1 + a_1, \qquad 0 \leq a_1 < b.$$

Vemos que a_1 é o segundo dígito à direita da expansão na base b de n. Continuamos este procedimento, dividindo sucessivamente os quocientes por b, obtendo dígitos adicionais na base b como restos. Este processo termina quando obtemos um quociente igual a zero.

EXEMPLO 3 Encontre a expansão na base 8, ou **octal**, de $(12345)_{10}$.

Exemplos Extras

Solução: Primeiro, divida 12345 por 8 para obter

$$12345 = 8 \cdot 1543 + 1.$$

Dividindo sucessivamente os quocientes por 8, temos

$$\begin{aligned} 1543 &= 8 \cdot 192 + 7, \\ 192 &= 8 \cdot 24 + 0, \\ 24 &= 8 \cdot 3 + 0, \\ 3 &= 8 \cdot 0 + 3. \end{aligned}$$

Como os restos são os dígitos da expansão na base 8 de 12345, temos que

$$(12345)_{10} = (30071)_8.$$ ◀

EXEMPLO 4 Encontre a expansão hexadecimal de $(177130)_{10}$.

Solução: Primeiro divida 177130 por 16 para obter

$$177130 = 16 \cdot 11070 + 10.$$

Dividindo sucessivamente os quocientes por 16, temos

$$\begin{aligned} 11\,070 &= 16 \cdot 691 + 14, \\ 691 &= 16 \cdot 43 + 3, \\ 43 &= 16 \cdot 2 + 11, \\ 2 &= 16 \cdot 0 + 2. \end{aligned}$$

Como os restos são os dígitos da expansão hexadecimal (base 16) de $(177\,130)_{10}$, temos que

$$(177\,130)_{10} = (2\text{B3EA})_{16}.$$

(Lembre-se de que os números inteiros 10, 11 e 14 correspondem aos dígitos hexadecimais A, B e E, respectivamente.) ◀

EXEMPLO 5 Encontre a expansão binária de $(241)_{10}$.

Solução: Primeiro divida 241 por 2 para obter

$$241 = 2 \cdot 120 + 1.$$

Dividindo sucessivamente os quocientes por 2, temos que

$$\begin{aligned} 120 &= 2 \cdot 60 + 0, \\ 60 &= 2 \cdot 30 + 0, \\ 30 &= 2 \cdot 15 + 0, \\ 15 &= 2 \cdot 7 + 1, \\ 7 &= 2 \cdot 3 + 1, \\ 3 &= 2 \cdot 1 + 1, \\ 1 &= 2 \cdot 0 + 1. \end{aligned}$$

Como os restos são os dígitos da expansão binária (base 2) de $(241)_{10}$, temos que

$$(241)_{10} = (1111\ 0001)_2.$$ ◀

O pseudocódigo dado no Algoritmo 1 encontra a expansão na base b $(a_{k-1} \ldots a_1 a_0)_b$ do número inteiro n (escrito na base 10).

ALGORITMO 1 Construção de Expansões na Base *b*.

procedure *expansão na base b* (n: inteiro positivo)
$q := n$
$k := 0$
while $q \neq 0$
begin
  $a_k := q$ **mod** b
  $q := \lfloor q/b \rfloor$
  $k := k + 1$
end {a expansão na base b de n é $(a_{k-1} \ldots a_1 a_0)_b$}

TABELA 1	Representação Hexadecimal, Octal e Binária dos Inteiros de 0 a 15.															
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Octal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Binária	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

No Algoritmo 1, q representa o quociente obtido pelas divisões sucessivas por b, começando com $q = n$. Os dígitos das expansões na base b são os restos dessas divisões e são dados por q **mod** b. O algoritmo termina quando um quociente $q = 0$ é alcançado.

Lembre-se: Note que o Algoritmo 1 pode ser pensado como um algoritmo voraz.

A conversão entre as expansões binária e hexadecimal é extremamente fácil, pois cada dígito hexadecimal corresponde a um bloco de quatro dígitos binários, sendo esta correspondência mostrada na Tabela 1 sem os 0s iniciais. (Deixamos a demonstração deste fato para os exercícios 11 e 12 no final desta seção.) Esta conversão é ilustrada no Exemplo 6.

EXEMPLO 6 Encontre a expansão hexadecimal de $(11\ 1110\ 1011\ 1100)_2$ e a expansão binária de $(A8D)_{16}$.

Solução: Para converter $(11\ 1110\ 1011\ 1100)_2$ em uma notação hexadecimal, agrupamos os dígitos binários em blocos de quatro, adicionando zeros iniciais no começo do bloco mais à esquerda, se necessário. Esses blocos são 0011, 1110, 1011 e 1100, que correspondem aos dígitos hexadecimais 3, E, B e C, respectivamente. Conseqüentemente, $(11\ 1110\ 1011\ 1100)_2 = (3EBC)_{16}$.

Para converter $(A8D)_{16}$ em notação binária, substituímos cada dígito hexadecimal por um bloco de quatro dígitos binários. Esses blocos são 1010, 1000, 1101. Conseqüentemente, $(A8D)_{16} = (1010\ 1000\ 1101)_2$. ◀

Algoritmos para Operações com Números Inteiros

Os algoritmos para a realização de operações com números inteiros usando suas expansões binárias são extremamente importantes em aritmética computacional. Descreveremos algoritmos para a adição e a multiplicação de dois números inteiros expressos em notação binária. Também analisaremos a complexidade computacional desses algoritmos, em termos do número atual de operações binárias usadas. Nesse estudo, suponha que as expansões binárias de a e b sejam

$$a = (a_{n-1}a_{n-2} \ldots a_1a_0)_2,\ b = (b_{n-1}b_{n-2} \ldots b_1b_0)_2,$$

para que a e b tenham n bits (colocando bits iguais a 0 no começo de uma dessas expansões, se necessário).

Mediremos a complexidade dos algoritmos para aritmética de números inteiros em termos do número de bits nesses números.

Considere o problema de somar dois números inteiros em notação binária. Um procedimento para realizar a adição pode ser baseado no método usual para adicionar números com papel e caneta. Este método consiste na adição de pares de dígitos binários juntos com transportes, quando eles aparecem, para computar a soma de dois números inteiros. Este procedimento será agora especificado em detalhes.

Para somar a e b, primeiro somamos seus bits mais à direita. Assim, temos

$$a_0 + b_0 = c_0 \cdot 2 + s_0,$$

em que s_0 é o bit mais à direita na expansão binária de $a + b$ e c_0 é o **transporte,** que é 0 ou 1. Então, adicione o próximo par de bits e o transporte,

$$a_1 + b_1 + c_0 = c_1 \cdot 2 + s_1,$$

em que s_1 é o próximo bit (da direita) na expansão binária de $a + b$, e c_1 é o transporte. Continue esse processo, adicionando os bits correspondentes nas duas expansões binárias e o transporte, para determinar o próximo bit sempre da direita para a esquerda na expansão binária de $a + b$. Na última etapa, adicione a_{n-1}, b_{n-1} e c_{n-2} para obter $c_{n-1} \cdot 2 + s_{n-1}$. O bit que termina a soma é $s_n = c_{n-1}$. Este procedimento fornece a expansão binária da soma, ou seja, $a + b = (s_n s_{n-1} s_{n-2} \ldots s_1 s_0)_2$.

EXEMPLO 7 Adicione $a = (1110)_2$ a $b = (1011)_2$.

Solução: Seguindo o procedimento especificado no algoritmo, primeiro note que

$$a_0 + b_0 = 0 + 1 = 0 \cdot 2 + 1,$$

em que $c_0 = 0$ e $s_0 = 1$. Então, como

$$a_1 + b_1 + c_0 = 1 + 1 + 0 = 1 \cdot 2 + 0,$$

temos que $c_1 = 1$ e $s_1 = 0$. Continuando,

$$a_2 + b_2 + c_1 = 1 + 0 + 1 = 1 \cdot 2 + 0,$$

em que $c_2 = 1$ e $s_2 = 0$. Por fim, como

$$a_3 + b_3 + c_2 = 1 + 1 + 1 = 1 \cdot 2 + 1,$$

temos que $c_3 = 1$ e $s_3 = 1$. Isto significa que $s_4 = c_3 = 1$. Assim, $s = a + b = (1\ 1001)_2$. Esta adição é mostrada na Figura 1. ◀

```
 1 1 1
   1 1 1 0
   1 0 1 1
 ---------
 1 1 0 0 1
```

FIGURA 1 **Adição de $(1110)_2$ a $(1011)_2$.**

O algoritmo de adição pode ser descrito usando-se o pseudocódigo, como vemos a seguir.

ALGORITMO 2 Adição de Números Inteiros.

procedure *adição*(*a*, *b*: integer inteiros positivos)
{as expansões binárias de a e b são $(a_{n-1}a_{n-2} \ldots a_1a_0)_2$
e $(b_{n-1}b_{n-2} \ldots b_1b_0)_2$, respectivamente}
$c := 0$
for $j := 0$ **to** $n - 1$
begin
  $d := \lfloor (a_j + b_j + c)/2 \rfloor$
  $s_j := a_j + b_j + c - 2d$
  $c := d$
end
$s_n := c$
{a expansão binária da soma é $(s_n s_{n-1} \ldots s_0)_2$}

A seguir, analisaremos o número de adições de bits usados pelo Algoritmo 2.

EXEMPLO 8 Quantas adições de bits são necessárias para usar o Algoritmo 2 para somar dois números inteiros com n bits (ou menos) em suas representações binárias?

Solução: Dois números inteiros são somados por sucessivas adições de pares de bits e um transporte, quando ocorrer. Adicionar cada par de bits e o transporte requer três ou menos adições de bits. Então, o número total de adições de bits usadas é menor que três vezes o número de bits na expansão. Assim, o número de adições de bits usadas pelo Algoritmo 2 para somar dois números inteiros com n bits é $O(n)$. ◀

Agora, considere a multiplicação de dois números inteiros com n bits a e b. O algoritmo convencional (usado quando multiplicamos com papel e caneta) funciona como podemos ver a seguir. Usando a lei distributiva, temos que

$$\begin{aligned} ab &= a(b_0 2^0 + b_1 2^1 + \cdots + b_{n-1} 2^{n-1}) \\ &= a(b_0 2^0) + a(b_1 2^1) + \cdots + a(b_{n-1} 2^{n-1}). \end{aligned}$$

Podemos computar ab usando essa equação. Primeiro notamos que $ab_j = a$, se $b_j = 1$, e $ab_j = 0$, se $b_j = 0$. Cada vez que multiplicamos um termo por 2, mudamos sua expansão binária uma casa para a esquerda e adicionamos zero ao final da expansão. Conseqüentemente, podemos obter $(ab_j)2^j$ pela **substituição** da expansão binária de ab_j j casas à esquerda e pela adição de j bits zero no final dessa expansão. Por fim, obtemos ab pela adição de n números inteiros $ab_j 2^j$, $j = 0, 1, 2, \ldots, n-1$.

Este procedimento para a multiplicação pode ser descrito usando-se o pseudocódigo, como mostrado no Algoritmo 3.

ALGORITMO 3 Multiplicação de Números Inteiros.

procedure *multiplicação* (a, b: inteiro)
{as expansões binárias de a e b são $(a_{n-1}a_{n-2}\ldots a_1a_0)_2$ e $(b_{n-1}b_{n-2}\ldots b_1b_0)_2$, respectivamente}
for $j := 0$ **to** $n-1$
begin
  if $b_j = 1$ **then** $c_j := a$ deslocado j lugares à esquerda
  else $c_j := 0$
end
{$c_0, c_1, \ldots, c_{n-1}$ são os produtos parciais}
$p := 0$
for $j := 0$ **to** $n-1$
  $p := p + c_j$
{p é o valor de ab}

O Exemplo 9 ilustra o uso desse algoritmo.

EXEMPLO 9 Encontre o produto de $a = (110)_2$ e $b = (101)_2$.

Solução: Primeiro note que

$$ab_0 \cdot 2^0 = (110)_2 \cdot 1 \cdot 2^0 = (110)_2,$$
$$ab_1 \cdot 2^1 = (110)_2 \cdot 0 \cdot 2^1 = (0000)_2,$$

```
  1 1 0
  1 0 1
  1 1 0
0 0 0
1 1 0
1 1 1 1 0
```

FIGURA 2 **Multiplicação de $(110)_2$ e $(101)_2$.**

e

$$ab_2 \cdot 2^2 = (110)_2 \cdot 1 \cdot 2^2 = (11000)_2.$$

Para encontrar o produto, adicione $(110)_2$, $(0000)_2$ e $(11000)_2$. Realizar essas adições (usando o Algoritmo 2, incluindo os bits zeros iniciais, quando necessário) mostra que $ab = (1\ 1110)_2$. Esta multiplicação é ilustrada na Figura 2. ◀

A seguir, determinamos o número de adições de bits e substituições de bits usadas pelo Algoritmo 3 para multiplicar dois números inteiros.

EXEMPLO 10 Quantas adições de bits e substituições de bits são usadas para multiplicar a e b usando o Algoritmo 3?

Solução: O Algoritmo 3 computa o produto de a e b pela adição de produtos parciais c_0, c_1, c_2, ... e c_{n-1}. Quando $b_j = 1$, computamos o produto parcial c_j pela substituição da expansão binária de a j bits. Quando $b_j = 0$, as substituições não são necessárias, porque $c_j = 0$. Assim, para encontrar todos os n números inteiros ab_j2^j, $j = 0, 1, \ldots, n-1$, é necessário, no máximo,

$$0 + 1 + 2 + \cdots + n - 1$$

substituições. Então, a partir do Exemplo 5 da Seção 3.2, o número de substituições necessárias é $O(n^2)$.

Para somar os números inteiros ab_j de $j = 0$ a $j = n-1$ é necessária a adição de um número inteiro com n bits, um inteiro com $(n+1)$ bits, ..., e um inteiro com $(2n)$ bits. Sabemos, a partir do Exemplo 8, que cada uma dessas adições requer $O(n)$ adições de bits. Conseqüentemente, um total de $O(n^2)$ adições de bits são necessárias para todas as n adições. ◀

ALGORITMO 4 Computar div e mod.

```
procedure algoritmo de divisao(a: número inteiro, d: número inteiro positivo)
q := 0
r := |a|
while r ≥ d
begin
    r := r − d
    q := q + 1
end
if a < 0 e r > 0 then
begin
    r := d − r
    q := −(q + 1)
end
{q = a div d é o quociente, r = a mod d é o resto}
```

Surpreendentemente, há algoritmos mais eficientes que o convencional para multiplicar números inteiros. Um desses algoritmos, que usa $O(n^{1.585})$ operações binárias para multiplicar números de n bits, será descrito na Seção 7.3.

Dados os números inteiros a e d, $d > 0$, podemos encontrar $q = a$ **div** d e $r = a$ **mod** d usando o Algoritmo 4. Neste algoritmo, quando a é positivo, subtraímos d de a quantas vezes forem

necessárias até que o resto seja menor que d. O número de vezes que realizamos essa subtração é o quociente, e o que sobrou de todas as subtrações é o resto. O Algoritmo 4 também é usado para o caso em que a é negativo. Ele encontra o quociente q e o resto r quando $|a|$ é dividido por d. Então, quando $a < 0$ e $r > 0$, ele os utiliza para encontrar o quociente $-(q + 1)$ e o resto $d - r$, quando a é dividido por d. Deixamos para o leitor (Exercício 57) mostrar que, supondo que $a > d$, este algoritmo usa $O(q \log a)$ operações binárias.

Há algoritmos mais eficientes que o Algoritmo 4 para determinar o quociente $q = a$ **div** d e o resto $r = a$ **mod** d quando um número inteiro positivo a é dividido por um número inteiro positivo d (veja [Kn98] para mais detalhes). Esses algoritmos necessitam de $O(\log a \cdot \log d)$ operações binárias. Se ambas as expansões binárias, a e d, contêm n ou menos bits, então podem substituir $\log a \cdot \log d$ por n^2. Isso significa que precisamos de $O(n^2)$ operações binárias para encontrar o quociente e o resto quando a é dividido por d.

Potenciação Modular

Em criptografia, é importante que saibamos como encontrar b^n **mod** m de maneira eficiente, em que b, n e m são números inteiros grandes. É impraticável computar primeiro b^n e depois encontrar seu resto quando dividido por m, pois b^n será um número muito grande. Em vez disso, podemos usar um algoritmo que emprega a expansão binária do expoente n, ou seja, $n = (a_{k-1} \ldots a_1 a_0)_2$.

Antes de apresentar esse algoritmo, ilustraremos sua idéia básica. Explicaremos como usar a expansão binária de n para computar b^n. Primeiro, note que

$$b^n = b^{a_{k-1} \cdot 2^{k-1} + \cdots + a_1 \cdot 2 + a_0} = b^{a_{k-1} \cdot 2^{k-1}} \cdots b^{a_1 \cdot 2} \cdot b^{a_0}.$$

Para computar b^n, encontramos os valores de b, b^2, $(b^2)^2 = b^4$, $(b^4)^2 = b^8$, $\ldots$, b^{2^k}. Depois, multiplicamos os termos b^{2^j} nesta lista, em que $a_j = 1$. Isto nos fornece b^n. Por exemplo, para computar 3^{11}, primeiro notamos que $11 = (1011)_2$, para que $3^{11} = 3^8 3^2 3^1$. Elevando ao quadrado sucessivamente, vemos que $3^2 = 9$, $3^4 = 9^2 = 81$, e $3^8 = (81)^2 = 6561$. Conseqüentemente, $3^{11} = 3^8 3^2 3^1 = 6561 \cdot 9 \cdot 3 = 177\,147$.

O algoritmo encontra sucessivamente b **mod** m, b^2 **mod** m, b^4 **mod** m, $\ldots$, $b^{2^{k-1}}$ **mod** m e multiplica esses termos junto com b^{2^j} **mod** m, em que $a_j = 1$, encontrando o resto do produto quando dividido por m depois de cada multiplicação. O pseudocódigo desse algoritmo é mostrado no Algoritmo 5. Note que no Algoritmo 5 usamos o algoritmo mais eficiente disponível para computar os valores da função **mod**, não o Algoritmo 4.

ALGORITMO 5 Potenciação Modular.

procedure *potenciacao modular* (b: número inteiro, $n = (a_{k-1}a_{k-2} \ldots a_1 a_0)_2$,
m: números inteiros positivos)
$x := 1$
potencia $:= b$ **mod** m
for $i := 0$ **to** $k - 1$
begin
  if $a_i = 1$ **then** $x := (x \cdot potencia)$ **mod** m
  potencia $:= (potencia \cdot potencia)$ **mod** m
end
{x é igual a b^n **mod** m}

Ilustramos agora, no Exemplo 11, como o Algoritmo 5 funciona.

EXEMPLO 11 Use o Algoritmo 5 para encontrar 3^{644} **mod** 645.

Solução: O Algoritmo 5 inicialmente designa $x = 1$ e *potencia* $= 3$ **mod** $645 = 3$. Na computação de 3^{644} **mod** 645, este algoritmo determina 3^{2^j} **mod** 645 para $j = 1, 2, \ldots, 9$ elevando ao quadrado sucessivamente e reduzindo pelo módulo 645. Se $a_j = 1$ (em que a_j é o bit na j-ésima posição na expansão binária de 644), multiplicamos o valor atual de x por 3^{2^j} **mod** 645 e reduzimos o resultado módulo 645. Aqui estão os passos percorridos:

$i = 0$: Como $a_0 = 0$, temos $x = 1$ e *potencia* $= 3^2$ **mod** $645 = 9$ **mod** $645 = 9$;

$i = 1$: Como $a_1 = 0$, temos $x = 1$ e *potencia* $= 9^2$ **mod** $645 = 81$ **mod** $645 = 81$;

$i = 2$: Como $a_2 = 1$, temos $x = 1 \cdot 81$ **mod** $645 = 81$ e *potencia* $= 81^2$ **mod** $645 = 6561$ **mod** $645 = 111$;

$i = 3$: Como $a_3 = 0$, temos $x = 81$ e *potencia* $= 111^2$ **mod** $645 = 12\,321$ **mod** $645 = 66$;

$i = 4$: Como $a_4 = 0$, temos $x = 81$ e *potencia* $= 66^2$ **mod** $645 = 4356$ **mod** $645 = 486$;

$i = 5$: Como $a_5 = 0$, temos $x = 81$ e *potencia* $= 486^2$ **mod** $645 = 236\,196$ **mod** $645 = 126$;

$i = 6$: Como $a_6 = 0$, temos $x = 81$ e *potencia* $= 126^2$ **mod** $645 = 15\,876$ **mod** $645 = 396$;

$i = 7$: Como $a_7 = 1$, vemos que $x = (81 \cdot 396)$ **mod** $645 = 471$ e *potencia* $= 396^2$ **mod** $645 = 156\,816$ **mod** $645 = 81$;

$i = 8$: Como $a_8 = 0$, temos $x = 471$ e *potencia* $= 81^2$ **mod** $645 = 6561$ **mod** $645 = 111$;

$i = 9$: Como $a_9 = 1$, vemos que $x = (471 \cdot 111)$ **mod** $645 = 36$.

Esses são os passos a serem seguidos pelo Algoritmo 5 para produzir o resultado 3^{644} **mod** $645 = 36$. ◀

O Algoritmo 5 é bem eficiente; ele usa $O((\log m)^2 \log n)$ operações binárias para encontrar b^n **mod** m (veja o Exercício 56).

O Algoritmo de Euclides

Links O método descrito na Seção 3.5 para computar o máximo divisor comum de dois números inteiros, usando as fatorações em números primos desses inteiros, é ineficiente. A razão é o tempo gasto para encontrar as fatorações em números primos. Estudaremos um método mais eficiente para encontrar o máximo divisor comum, chamado de **algoritmo de Euclides.** Este algoritmo é conhecido desde tempos remotos. Foi assim nomeado em homenagem ao matemático grego Euclides, que incluiu sua descrição no livro *Os Elementos.*

Antes de descrever o algoritmo de Euclides, mostraremos como ele é usado para encontrar o mdc(91, 287). Primeiro, dividimos 287, o maior dos dois inteiros, por 91, o menor, e obtemos

$$287 = 91 \cdot 3 + 14.$$

Qualquer divisor de 91 e 287 deve também ser divisor de $287 - 91 \cdot 3 = 14$. Além disso, qualquer divisor de 91 e 14 deverá ser também divisor de $287 = 91 \cdot 3 + 14$. Assim, o máximo divisor comum de 91 e 287 é o mesmo que o máximo divisor comum de 91 e 14. Isto significa que o problema de encontrar o mdc(91, 287) foi reduzido ao problema de encontrar o mdc(91, 14).

Podemos, então, prosseguir com as divisões sucessivas. Dividimos 91 por 14 para obter

$$91 = 14 \cdot 6 + 7.$$

Como qualquer divisor comum de 91 e de 14 também divide $91 - 14 \cdot 6 = 7$ e qualquer divisor comum de 14 e de 7 divide 91, temos que $\text{mdc}(91, 14) = \text{mdc}(14, 7)$.

Continuando com a divisão de 14 por 7, obtemos

$$14 = 7 \cdot 2.$$

Como 7 divide 14, temos que $\text{mdc}(14, 7) = 7$. Além disso, como $\text{mdc}(287, 91) = \text{mdc}(91, 14) = \text{mdc}(14, 7) = 7$, o problema original foi resolvido.

Descreveremos agora como o algoritmo de Euclides funciona em questões mais gerais. Usaremos divisões sucessivas para reduzir o problema de encontrar o máximo divisor comum de dois números inteiros positivos no mesmo problema com números inteiros menores, até um dos inteiros chegar a zero.

O algoritmo de Euclides é baseado no resultado a seguir sobre máximos divisores comuns e o algoritmo de divisão.

LEMA 1 Considere $a = bq + r$, em que a, b, q e r são números inteiros. Então, $\text{mdc}(a, b) = \text{mdc}(b, r)$.

Demonstração: Se pudermos mostrar que os divisores comuns de a e b são os mesmos que os divisores comuns de b e r, teremos mostrado que $\text{mdc}(a, b) = \text{mdc}(b, r)$, pois ambos os pares devem ter o mesmo *máximo* divisor comum.

Então, suponha que d divida ambos, a e b. Assim, temos que d também divide $a - bq = r$ (a partir do Teorema 1 da Seção 3.4). Dessa maneira, qualquer divisor comum de a e b é também divisor comum de b e r.

Da mesma forma, suponha que d divida ambos, b e r. Então, d também divide $bq + r = a$. Assim, qualquer divisor comum de b e r é também divisor comum de a e b.

Conseqüentemente, $\text{mdc}(a, b) = \text{mdc}(b, r)$. ◁

Suponha que a e b sejam números inteiros positivos com $a \geq b$. Considere $r_0 = a$ e $r_1 = b$. Quando aplicamos sucessivamente o algoritmo de divisão, obtemos

$$\begin{aligned}
r_0 &= r_1 q_1 + r_2 && 0 \leq r_2 < r_1,\\
r_1 &= r_2 q_2 + r_3 && 0 \leq r_3 < r_2,\\
&\ \ \vdots\\
r_{n-2} &= r_{n-1} q_{n-1} + r_n && 0 \leq r_n < r_{n-1},\\
r_{n-1} &= r_n q_n.
\end{aligned}$$

Sempre aparecerá um resto zero nesta seqüência de divisões sucessivas, pois a seqüência de restos $a = r_0 > r_1 > r_2 > \cdots \geq 0$ não pode ter mais que a termos. Além disso, temos, a partir do Lema 1, que

$$\begin{aligned}
\text{mdc}(a, b) &= \text{mdc}(r_0, r_1) = \text{mdc}(r_1, r_2) = \cdots = \text{mdc}(r_{n-2}, r_{n-1})\\
&= \text{mdc}(r_{n-1}, r_n) = \text{mdc}(r_n, 0) = r_n.
\end{aligned}$$

Links Assim, o máximo divisor comum é o ultimo resto diferente de zero na seqüência de divisões.

EUCLIDES (325–265 a.C.) Euclides foi o autor do livro de matemática de maior sucesso já escrito, *Os Elementos*, que foi editado em mais de 1000 edições diferentes desde a antigüidade até os tempos modernos. A vida de Euclides é pouco conhecida, mas sabe-se que ele ensinou em uma famosa academia em Alexandria, no Egito. Aparentemente, Euclides não enfatizou as aplicações. Quando um estudante perguntou o que poderia ganhar aprendendo geometria, Euclides explicou que o valor do conhecimento adquirido era para seu próprio enriquecimento, e pediu a seu servo que desse uma moeda ao estudante "porque ele precisa ter lucro com aquilo que aprende".

EXEMPLO 12 Encontre o máximo divisor comum de 414 e 662 usando o algoritmo de Euclides.

Solução: Usando sucessivamente o algoritmo de divisão, temos:

$$662 = 414 \cdot 1 + 248$$
$$414 = 248 \cdot 1 + 166$$
$$248 = 166 \cdot 1 + 82$$
$$166 = 82 \cdot 2 + 2$$
$$82 = 2 \cdot 41.$$

Assim, mdc(414, 662) = 2, pois 2 é o último resto diferente de zero. ◀

O algoritmo de Euclides é expresso em pseudocódigo no Algoritmo 6.

ALGORITMO 6 O Algoritmo de Euclides.

procedure mdc(a, b: números inteiros positivos)
$x := a$
$y := b$
while $y \neq 0$
begin
    $r := x$ **mod** y
    $x := y$
    $y := r$
end {mdc(a, b) é x}

No Algoritmo 6, os valores iniciais de x e y são a e b, respectivamente. Em cada etapa do procedimento, x é substituído por y, e y é substituído por x **mod** y, que é o resto da divisão de x por y. Este processo é repetido até que $y \neq 0$. O algoritmo termina quando $y = 0$ e o valor de x neste ponto, o último resto diferente de zero, é o máximo divisor comum de a e b.

Estudaremos a complexidade temporal do algoritmo de Euclides na Seção 4.3, na qual mostraremos que o número de divisões necessárias para encontrar o máximo divisor comum de a e b, em que $a \geq b$, é $O(\log b)$.

Exercícios

1. Converta estes números inteiros da notação decimal para a binária.
   a) 231 **b)** 4 532 **c)** 97 644
2. Converta estes números inteiros da notação decimal para a binária.
   a) 321 **b)** 1 023 **c)** 100 632
3. Converta estes números inteiros da notação binária para a decimal.
   a) 1 1111 **b)** 10 0000 0001
   c) 1 0101 0101 **d)** 110 1001 0001 0000
4. Converta estes números inteiros da notação binária para a decimal.
   a) 1 1011 **b)** 10 1011 0101
   c) 11 1011 1110 **d)** 111 1100 0001 1111
5. Converta estes números inteiros da notação hexadecimal para a binária.
   a) 80E **b)** 135AB
   c) ABBA **d)** DEFACED
6. Converta $(\text{BADFACED})_{16}$ da expansão hexadecimal para a binária.

7. Converta $(ABCDEF)_{16}$ da expansão hexadecimal para a binária.

8. Converta cada um destes números inteiros da notação binária para a hexadecimal.
a) 1111 0111
b) 1010 1010 1010
c) 111 0111 0111 0111

9. Converta $(1011\ 0111\ 1011)_2$ da expansão binária para a hexadecimal.

10. Converta $(1\ 1000\ 0110\ 0011)_2$ da expansão binária para a hexadecimal.

11. Mostre que a expansão hexadecimal de um número inteiro positivo pode ser obtida a partir de sua expansão binária a partir do agrupamento de blocos de quatro dígitos binários, adicionando dígitos iniciais, se necessário, e transcrevendo cada bloco de quatro dígitos binários em um único dígito hexadecimal.

12. Mostre que a expansão binária de um número inteiro positivo pode ser obtida a partir de sua expansão hexadecimal transcrevendo cada dígito hexadecimal em um bloco de quatro dígitos binários.

13. Dê um procedimento simples para converter uma expansão binária de um número inteiro em uma expansão octal.

14. Dê um procedimento simples para converter uma expansão octal de um número inteiro em uma expansão binária.

15. Converta $(7345321)_8$ para sua expansão binária e $(10\ 1011\ 1011)_2$ para sua expansão octal.

16. Dê um procedimento para converter uma expansão hexadecimal de um número inteiro em sua expansão octal usando a notação binária como etapa intermediária.

17. Dê um procedimento para converter uma expansão octal de um número inteiro em sua expansão hexadecimal usando a notação binária como etapa intermediária.

18. Converta $(12345670)_8$ em sua expansão hexadecimal e $(ABB093BABBA)_{16}$ em sua expansão octal.

19. Use o Algoritmo 5 para encontrar 7^{644} **mod** 645.

20. Use o Algoritmo 5 para encontrar 11^{644} **mod** 645.

21. Use o Algoritmo 5 para encontrar 3^{2003} **mod** 99.

22. Use o Algoritmo 5 para encontrar 123^{1001} **mod** 101.

23. Use o algoritmo de Euclides para encontrar
a) mdc(12, 18). **b)** mdc(111, 201).
c) mdc(1001, 1331). **d)** mdc(12345, 54321).
e) mdc(1000, 5040). **f)** mdc(9888, 6060).

24. Use o algoritmo de Euclides para encontrar
a) mdc(1, 5). **b)** mdc(100, 101).
c) mdc(123, 277). **d)** mdc(1529, 14039).
e) mdc(1529, 14038). **f)** mdc(11111, 111111).

25. Quantas divisões são necessárias para encontrar mdc(21, 34) usando o algoritmo de Euclides?

26. Quantas divisões são necessárias para encontrar mdc(34, 55) usando o algoritmo de Euclides?

27. Mostre que todo número inteiro positivo pode ser representado unicamente como a soma de distintas potências de 2. [*Dica:* Considere as expansões binárias de números inteiros.]

28. Podemos mostrar que todo número inteiro pode ser representado unicamente na forma

$$e_k 3^k + e_{k-1} 3^{k-1} + \cdots + e_1 3 + e_0,$$

em que $e_j = -1$, 0 ou 1 para $j = 0, 1, 2, \ldots, k$. As expansões deste tipo são chamadas de **expansões ternárias balanceadas.** Encontre as expansões ternárias balanceadas de
a) 5. **b)** 13. **c)** 37. **d)** 79.

29. Mostre que um número inteiro positivo é divisível por 3 se e somente se a soma de seus dígitos decimais for divisível por 3.

30. Mostre que um número inteiro positivo é divisível por 11 se e somente se a diferença da soma dos dígitos decimais em posições pares e a soma dos seus dígitos decimais em posições ímpares for divisível por 11.

31. Mostre que um número inteiro positivo é divisível por 3 se e somente se a diferença da soma de seus dígitos binários em posições pares e a soma de seus dígitos binários em posições ímpares for divisível por 3.

As representações **complemento de um** de números inteiros são usadas para simplificar a aritmética da computação. Para representar números inteiros positivos e negativos com valores absolutos menores que 2^{n-1}, é usado um total de n bits. O bit na extremidade mais à esquerda é usado para representar o sinal. Um bit 0 nesta posição é usado para números inteiros positivos, e um bit 1 nesta posição é usado para números inteiros negativos. Para números inteiros positivos, os bits restantes são idênticos à expansão binária do inteiro. Para números inteiros negativos, os bits restantes são obtidos encontrando-se primeiro a expansão binária de valor absoluto do inteiro e, então, tomando-se o complemento de cada um desses bits, em que o complemento de 1 é 0, e o complemento de 0 é 1.

32. Encontre as representações complemento de um, usando as cadeias de bits de extensão seis dos seguintes números inteiros:
a) 22 **b)** 31 **c)** −7 **d)** −19

33. Qual número inteiro representa cada uma das representações complemento de um de extensão cinco?
a) 11001 **b)** 01101 **c)** 10001 **d)** 11111

34. Se m é um número inteiro positive menor que 2^{n-1}, como a representação complemento de um de $-m$ é obtida a partir da representação complemento de um de m, quando as cadeias de bits de extensão n são usadas?

35. Como a representação complemento de um da soma de dois números inteiros é obtida a partir das representações complemento de um desses inteiros?

36. Como a representação complemento de um da diferença de dois números inteiros é obtida a partir das representações complemento de um desses inteiros?

37. Mostre que o número inteiro m com representação complemento de um $(a_{n-1}a_{n-2} \ldots a_1 a_0)$ pode ser encontrada usando-se a equação $m = -a_{n-1}(2^{n-1} - 1) + a_{n-2}2^{n-2} + \cdots + a_1 \cdot 2 + a_0$.

As representações **complemento de dois** de números inteiros também são usadas para simplificar a aritmética da computação e são mais usadas que as representações complemento de um. Para representar um número inteiro x com $-2^{n-1} \leq x \leq 2^{n-1} - 1$ para um número inteiro positivo específico n, é usado um total de n bits. O bit

mais à esquerda é usado para representar o sinal. Um bit 0 nesta posição é usado para números inteiros positivos, e um bit 1 nesta posição é usado para números inteiros negativos, como nas expansões de complemento de um. Para um número inteiro positivo, os bits restantes são idênticos para a expansão binária do inteiro. Para um número inteiro negativo, os bits restantes são os bits da expansão binária de $2^{n-1} - |x|$. As expansões complemento de dois de inteiros são geralmente usadas por computadores porque a adição e a subtração de números inteiros podem ser realizadas facilmente usando essas expansões, em que tais inteiros podem ser positivos ou negativos.

38. Responda ao Exercício 32, mas desta vez encontre a expansão complemento de dois usando as cadeias de bits de extensão seis.

39. Responda ao Exercício 33, considerando que cada expansão é uma expansão complemento de dois de extensão cinco.

40. Responda ao Exercício 34 para expansões complemento de dois.

41. Responda ao Exercício 35 para expansões complemento de dois.

42. Responda ao Exercício 36 para expansões complemento de dois.

43. Mostre que o número inteiro m com representação complemento de dois $(a_{n-1}a_{n-2} \ldots a_1a_0)$ pode ser encontrado usando-se a equação $m = -a_{n-1} \cdot 2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_1 \cdot 2 + a_0$.

44. Dê um algoritmo simples para formar a representação complemento de dois de um número inteiro a partir de sua representação complemento de um.

45. Às vezes, números inteiros são codificados usando-se expansões binárias de quatro dígitos para representar cada dígito decimal. Isto produz a forma **codificação binária decimal** do inteiro. Por exemplo, 791 é codificado desta maneira para 011110010001. Quantos bits são necessários para representar um número com n dígitos decimais usando este tipo de codificação?

Uma **expansão de Cantor** é uma soma na forma

$$a_n n! + a_{n-1}(n-1)! + \cdots + a_2 2! + a_1 1!,$$

em que a_i é um número inteiro com $0 \leq a_i \leq i$ para $i = 1, 2, \ldots, n$.

46. Encontre a expansão de Cantor de

a) 2. **b)** 7. **c)** 19.
d) 87. **e)** 1000. **f)** 1 000 000.

***47.** Descreva um algoritmo para encontrar a expansão de Cantor de um número inteiro.

***48.** Descreva um algoritmo para somar dois números inteiros a partir de suas expansões de Cantor.

49. Some $(10111)_2$ e $(11010)_2$ fazendo cada etapa do algoritmo de adição dado no texto.

50. Multiplique $(1110)_2$ por $(1010)_2$ fazendo cada etapa do algoritmo de multiplicação dado no texto.

51. Descreva um algoritmo para encontrar a diferença de dois números em suas expansões binárias.

52. Estime o número de operações binárias usadas para subtrair duas expansões binárias.

53. Construa um algoritmo que, dadas as expansões binárias dos inteiros a e b, determine se $a > b$, $a = b$ ou $a < b$.

54. Quantas operações binárias são usadas para comparar os números no algoritmo do Exercício 53 quando o maior de a e b tem n bits em sua expansão binária?

55. Estime a complexidade do Algoritmo 1 para encontrar a expansão de base b de um número inteiro n em termos do número de divisões usadas.

***56.** Mostre que o Algoritmo 5 usa $O((\log m)^2 \log n)$ operações binárias para encontrar b^n **mod** m.

57. Mostre que o Algoritmo 4 usa $O(q \log a)$ operações binárias, supondo que $a > d$.

3.7 Aplicações da Teoria dos Números

Introdução

A teoria dos números tem muitas aplicações, especialmente na ciência computacional. Na Seção 3.4, descrevemos muitas dessas aplicações, incluindo as funções de hashing, a construção de números pseudo-aleatórios e os códigos de substituição. Nesta seção continuaremos com nossa introdução à teoria dos números, desenvolvendo alguns resultados-chave e apresentando duas aplicações importantes: um método para a realização da aritmética com números inteiros grandes e um tipo de sistema de criptografia criado recentemente, chamado de *sistema de chave pública*. Em tal sistema, não é necessário manter chaves secretas codificadas, porque o conhecimento de uma chave desse tipo não ajuda na decodificação de mensagens em tempo real. Chaves de decodificação privadas são usadas para decodificar mensagens.

Antes de desenvolver essas aplicações, apresentaremos alguns resultados-chave que têm um papel central na teoria dos números e em suas aplicações. Por exemplo, mostraremos como resolver um sistema de congruência linear modular de pares de primos entre si usando o Teorema Chinês do Resto e, então, mostraremos como usar esse resultado como base para a realização da aritmética com números inteiros grandes. Discutiremos o Pequeno Teorema de Fermat e o conceito de um pseudoprimo e mostraremos como usar esses conceitos para desenvolver um criptossistema de chave pública.

Alguns Resultados Úteis

Um resultado importante que usaremos nesta seção é que o máximo divisor comum de dois números inteiros a e b pode ser expresso na forma

$$sa + tb,$$

em que s e t são números inteiros. Em outras palavras, mdc(a, b) pode ser expresso como uma **combinação linear** com coeficientes inteiros de a e b. Por exemplo, mdc$(6, 14) = 2$, e $2 = (-2) \cdot 6 + 1 \cdot 14$. Afirmamos este fato como Teorema 1.

TEOREMA 1 Se a e b são números inteiros positivos, então existem inteiros s e t tal que mdc$(a, b) = sa + tb$.

Não faremos uma demonstração formal do Teorema 1 aqui (para isso, veja o Exercício 36 na Seção 4.2 e também [Ro05]), mas vamos fornecer o exemplo de um método para encontrar a combinação linear de dois números inteiros igual ao seu máximo divisor comum. (Nesta seção, vamos supor que uma combinação linear tem coeficientes inteiros.) O método consiste em trabalhar em sentido inverso as divisões do algoritmo de Euclides. (Também descrevemos, no preâmbulo do Exercício 48, um algoritmo chamado de **algoritmo de Euclides estendido**, que pode ser usado para expressar mdc(a, b) como uma combinação linear de a e b.)

EXEMPLO 1 Expresse o mdc$(252, 198) = 18$ como uma combinação linear de 252 e 198.

Solução: Para mostrar que mdc$(252, 198) = 18$, o algoritmo de Euclides usa as seguintes divisões:

$$\begin{aligned} 252 &= 1 \cdot 198 + 54 \\ 198 &= 3 \cdot 54 + 36 \\ 54 &= 1 \cdot 36 + 18 \\ 36 &= 2 \cdot 18. \end{aligned}$$

Vemos, pela terceira divisão, que podemos expressar mdc$(252, 198) = 18$ como uma combinação linear de 54 e 36. Verificamos que

$$18 = 54 - 1 \cdot 36.$$

A segunda divisão nos mostra que

$$36 = 198 - 3 \cdot 54.$$

Substituindo essa expressão por 36 na equação anterior, podemos expressar 18 como uma combinação linear de 54 e 198. Temos

$$18 = 54 - 1 \cdot 36 = 54 - 1 \cdot (198 - 3 \cdot 54) = 4 \cdot 54 - 1 \cdot 198.$$

A primeira divisão nos mostra que

$$54 = 252 - 1 \cdot 198.$$

Substituindo essa expressão por 54 na equação anterior, podemos expressar 18 como uma combinação linear de 252 e 198. Concluímos que

$$18 = 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 = 4 \cdot 252 - 5 \cdot 198,$$

o que completa a solução. ◀

Usaremos o Teorema 1 para produzir muitos resultados úteis. Um de nossos objetivos será demonstrar parte do Teorema Fundamental da Aritmética, que diz que um número inteiro positivo tem, pelo menos, uma fatoração em números primos. Mostraremos que se um número inteiro positivo tem uma fatoração em números primos, em que os primos são escritos em ordem não decrescente, então esta fatoração é única.

Primeiro, precisamos desenvolver alguns resultados sobre divisibilidade.

LEMA 1 Se a, b e c são números inteiros positivos, tal que $\text{mdc}(a, b) = 1$ e $a \mid bc$, então $a \mid c$.

Demonstração: Como $\text{mdc}(a, b) = 1$, pelo Teorema 1, existem números inteiros s e t, tal que

$$sa + tb = 1.$$

Multiplicando os dois lados desta equação por c, obtemos

$$sac + tbc = c.$$

Usando o Teorema 1 da Seção 3.4, podemos usar esta última equação para mostrar que $a \mid c$. Pelo item (*ii*) daquele teorema, $a \mid tbc$. Como $a \mid sac$ e $a \mid tbc$, a partir do item (*i*) daquele mesmo teorema, concluímos que a divide $sac + tbc$ e, assim, $a \mid c$. Isto finaliza a demonstração. ◁

Utilizaremos a seguinte generalização do Lema 1 na demonstração de unicidade da fatoração em números primos. (A demonstração do Lema 2 será apresentada no Exercício 60 na Seção 4.1, porque ela pode ser mais bem trabalhada usando-se o método de indução matemática, que será abordado naquela seção.)

LEMA 2 Se p é um número primo e $p \mid a_1 a_2 \cdots a_n$, em que cada a_i é um número inteiro, então $p \mid a_i$ para algum i.

Mostraremos agora que a fatoração de um número inteiro em primos é única. Ou seja, mostraremos que todo número inteiro pode ser escrito como o produto de números primos em ordem não decrescente no máximo de uma maneira. Isto é parte do Teorema Fundamental da Aritmética. Demonstraremos a outra parte, que todo número inteiro pode ser fatorado em números primos, na Seção 4.2.

Demonstração (da unicidade da fatoração em primos de um número inteiro positivo): Utilizaremos uma demonstração por contradição. Suponha que o número inteiro positivo n possa ser escrito como o produto de números primos de duas maneiras diferentes, digamos, $n = p_1 p_2 \cdots p_s$ e $n = q_1 q_2 \cdots q_t$, sendo que cada p_i e q_j são números primos, tal que $p_1 \leq p_2 \leq \cdots \leq p_s$ e $q_1 \leq q_2 \leq \cdots \leq q_t$.

Quando retiramos todos os primos comuns das duas fatorações, temos

$$p_{i_1} p_{i_2} \cdots p_{i_u} = q_{j_1} q_{j_2} \cdots q_{j_v},$$

em que nenhum primo aparece em nenhum dos lados dessa equação, ou seja, todos os fatores são diferentes e u e v são números inteiros positivos. Pelo Lema 1, temos que p_{i_1} divide q_{j_k}, para algum k. Como nenhum primo divide outro número primo, isto é impossível. Conseqüentemente, pode existir no máximo uma fatoração de n em números primos em ordem não decrescente. ◁

O Lema 1 também pode ser usado para demonstrar um resultado da divisão dos dois lados de uma congruência pelo mesmo número inteiro. Mostramos (Teorema 5 da Seção 3.4) que podemos multiplicar os dois lados de uma congruência pelo mesmo número inteiro. Entretanto, dividir os dois lados de uma congruência por um número inteiro não produz sempre uma congruência válida, como mostra o Exemplo 2.

EXEMPLO 2 A congruência $14 \equiv 8 \pmod 6$ é válida, mas os dois lados dessa congruência não podem ser divididos por 2, pois $14/2 = 7$ e $8/2 = 4$, mas $7 \not\equiv 4 \pmod 6$. ◀

Entretanto, usando o Lema 1, mostramos que podemos dividir os dois lados de uma congruência por um número inteiro relativamente primo ao módulo. Isto é afirmado como Teorema 2.

TEOREMA 2 Considere m como um número inteiro positivo e a, b e c como números inteiros. Se $ac \equiv bc \pmod m$ e $\text{mdc}(c,m) = 1$, então $a \equiv b \pmod m$.

Demonstração: Como $ac \equiv bc \pmod m$, $m \mid ac - bc = c(a - b)$. Pelo Lema 1, como $\text{mdc}(c,m) = 1$, temos que $m \mid a - b$. E concluímos que $a \equiv b \pmod m$. ◁

Congruências Lineares

Uma congruência na forma

$$ax \equiv b \pmod m,$$

em que m é um número inteiro positivo, a e b são números inteiros e x é uma variável, é chamada de **congruência linear.** Tais congruências aparecem na teoria dos números e em suas aplicações.

Como podemos resolver a congruência linear $ax \equiv b \pmod m$, ou seja, encontrar todos os números inteiros x que satisfazem essa congruência? O método que vamos descrever usa um número inteiro $\overline{a}$, tal que $\overline{a}a \equiv 1 \pmod m$, se tal número inteiro existir. Dizemos que tal inteiro $\overline{a}$ é **inverso** de a módulo m. O Teorema 3 garante que um inverso de a modulo m existe sempre que a e m forem primos entre si.

TEOREMA 3 Se a e m são inteiros primos entre si e $m > 1$, então existe um inverso de a módulo m. Além disso, esse inverso é o único módulo m. (Ou seja, há um único número inteiro positivo $\overline{a}$ menor que m que é um inverso de a módulo m e todo outro inverso de a módulo m é congruente a $\overline{a}$ módulo m.)

Demonstração: Pelo Teorema 1, como $\text{mdc}(a,m) = 1$, existem números inteiros s e t , tal que

$$sa + tm = 1.$$

Isto implica que

$$sa + tm \equiv 1 \pmod m.$$

Como $tm \equiv 0 \pmod m$, temos que

$$sa \equiv 1 \pmod m.$$

Conseqüentemente, s é um inverso de a módulo m. Verificaremos, no Exercício 9, no final desta seção, que este inverso é o único módulo m. ◁

A demonstração do Teorema 3 descreve um método para encontrar o inverso de a módulo m, quando a e m são primos entre si: encontre uma combinação linear de a e m que seja igual a 1 (ela pode ser feita trabalhando de trás para frente os passos do algoritmo de Euclides); o coeficiente de a nesta combinação linear é um inverso de a módulo m. Ilustramos este procedimento no Exemplo 3.

EXEMPLO 3 Encontre um inverso de 3 módulo 7.

Solução: Como mdc(3, 7) = 1, o Teorema 3 nos diz que um inverso de 3 módulo 7 existe. O algoritmo de Euclides é rápido quando usado para encontrar o máximo divisor comum de 3 e 7:

$$7 = 2 \cdot 3 + 1.$$

A partir dessa equação, vemos que

$$-2 \cdot 3 + 1 \cdot 7 = 1.$$

Isso mostra que -2 é um inverso de 3 módulo 7. (Note que todo número inteiro congruente a -2 módulo 7 é também um inverso de 3, como 5, -9, 12, e assim por diante.) ◀

Quando temos um inverso $\bar{a}$ de a módulo m, podemos facilmente resolver a congruência $ax \equiv b \pmod{m}$ multiplicando os dois lados da congruência linear por $\bar{a}$, como o Exemplo 4 ilustra.

EXEMPLO 4 Quais são as soluções da congruência linear $3x \equiv 4 \pmod{7}$?

Solução: Pelo Exemplo 3, sabemos que -2 é um inverso de 3 módulo 7. Multiplicando os dois lados da congruência por -2 , temos que

$$-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}.$$

Como $-6 \equiv 1 \pmod{7}$ e $-8 \equiv 6 \pmod{7}$, temos que se x for uma solução, então $x \equiv -8 \equiv 6 \pmod{7}$.

Precisamos determinar se todo x com $x \equiv 6 \pmod{7}$ é uma solução. Suponha que $x \equiv 6 \pmod{7}$. Então, pelo Teorema 5 da Seção 3.4, temos que

$$3x \equiv 3 \cdot 6 = 18 \equiv 4 \pmod{7},$$

que mostra que todo x satisfaz a congruência. Concluímos que as soluções para a congruência são os números inteiros x , tal que $x \equiv 6 \pmod{7}$, ou seja, 6, 13, 20, . . . e -1, -8,-15, . . . ◀

O Teorema Chinês do Resto

Os sistemas de congruências lineares aparecem em muitos contextos. Por exemplo, como vimos anteriormente, eles são a base para um método que pode ser usado para realizar a aritmética com números inteiros grandes. Tais sistemas podem até ser encontrados como enigmas nos escritos dos antigos matemáticos chineses e hindus, como mostra o Exemplo 5.

Links

EXEMPLO 5 No primeiro século, o matemático chinês Sun-Tzu perguntou:

> Há certas coisas cuja quantidade é desconhecida. Quando dividida por 3, o resto é 2; quando dividida por 5, o resto é 3; e quando dividida por 7, o resto é 2. Qual será a quantidade dessas coisas?

Este enigma pode ser transcrito como a seguinte questão: Quais são as soluções para os sistemas de congruência

$$x \equiv 2 \pmod 3,$$
$$x \equiv 3 \pmod 5,$$
$$x \equiv 2 \pmod 7?$$

Resolveremos esse sistema e o enigma de Sun-Tzu ainda nesta seção. ◀

O *Teorema Chinês do Resto*, que recebe este nome em homenagem à herança chinesa de abordar problemas que envolvem sistemas de congruência linear, afirma que quando os módulos de um sistema de congruência linear são pares de primos entre si, há uma única solução do sistema módulo produto dos módulos.

TEOREMA 4

O TEOREMA CHINÊS DO RESTO Considere $m_1, m_2, \ldots, m_n$ números inteiros positivos de pares de primos entre si e $a_1, a_2, \ldots, a_n$ números inteiros arbitrários. Então, o sistema

$$x \equiv a_1 \pmod{m_1},$$
$$x \equiv a_2 \pmod{m_2},$$
$$\vdots$$
$$x \equiv a_n \pmod{m_n}$$

tem uma única solução módulo $m = m_1 m_2 \cdots m_n$. (Ou seja, há uma solução x com $0 \leq x < m$, e todas as outras soluções são congruentes módulo m a esta solução.)

Demonstração: Para estabelecer esse teorema, precisamos mostrar que uma solução existe e que ela é única módulo m. Vamos mostrar que há uma solução que descreve um modo de construir essa solução, processo que está ilustrado no Exercício 24, no final desta seção.

Para construir uma solução simultânea, primeiro considere

$$M_k = m/m_k$$

para $k = 1, 2, \ldots, n$. Ou seja, M_k é o produto dos módulos, exceto para m_k. Como m_i e m_k não têm fatores comuns maiores que 1, quando $i \neq k$, temos que $\text{mdc}(m_k, M_k) = 1$. Conseqüentemente, pelo Teorema 3, sabemos que há um número inteiro y_k, um inverso de M_k módulo m_k, tal que

$$M_k y_k \equiv 1 \pmod{m_k}.$$

Para construir uma solução simultânea, faça a soma

$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_n M_n y_n.$$

Vamos mostrar agora que x é uma solução simultânea. Primeiro, note que como $M_j \equiv 0 \pmod{m_k}$ sempre que $j \neq k$, todos os termos, exceto o k-ésimo desta soma, são congruentes a 0 módulo m_k. Como $M_k y_k \equiv 1 \pmod{m_k}$, vemos que

$$x \equiv a_k M_k y_k \equiv a_k \pmod{m_k},$$

para $k = 1, 2, \ldots, n$. Mostramos que x é uma solução simultânea às n congruências. ◁

O Exemplo 6 ilustra como usar a construção dada na demonstração do Teorema 4 para resolver um sistema de congruências. Resolveremos o sistema dado no Exemplo 5, chegando até o enigma de Sun-Tzu.

EXEMPLO 6 Para resolver o sistema de congruências do Exemplo 5, primeiro considere $m = 3 \cdot 5 \cdot 7 = 105$, $M_1 = m/3 = 35$, $M_2 = m/5 = 21$ e $M_3 = m/7 = 15$. Vemos que 2 é um inverso de $M_1 = 35$ módulo 3, porque $35 \equiv 2 \pmod 3$; 1 é um inverso de $M_2 = 21$ módulo 5, porque $21 \equiv 1 \pmod 5$; e 1 é um inverso de $M_3 = 15 \pmod 7$, porque $15 \equiv 1 \pmod 7$. As soluções desse sistema são aqueles x, tal que

$$x \equiv a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 = 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 = 233 \equiv 23 \pmod{105}.$$

Temos que 23 é o menor número inteiro positivo que é uma solução simultânea. Concluímos que 23 é o menor número inteiro positivo que tem resto 2 quando divido por 3, um resto 3 quando dividido por 5 e um resto 2 quando dividido por 7. ◀

Aritmética da Computação para Números Inteiros Grandes

Suponha que $m_1, m_2, \ldots, m_n$ sejam pares de primos entre si e inteiros maiores que ou iguais a 2 e considere m como seu produto. Pelo Teorema Chinês do Resto, podemos mostrar (veja o Exercício 22) que um número inteiro a com $0 \leq a < m$ pode ser representado unicamente pela n-upla que consiste dos restos da divisão por m_i, $i = 1, 2, \ldots, n$. Ou seja, podemos representar unicamente a por

$$(a \bmod m_1, a \bmod m_2, \ldots, a \bmod m_n).$$

EXEMPLO 7 Quais são os pares usados para representar os números inteiros não negativos menores que 12 quando eles são representados pelo par ordenado no qual o primeiro componente é o resto da divisão do número inteiro por 3 e o segundo componente é o resto da divisão do número inteiro por 4?

Solução: Temos as seguintes representações, obtidas pelo resto de cada número inteiro dividido por 3 e por 4:

$0 = (0, 0)$ $4 = (1, 0)$ $8 = (2, 0)$
$1 = (1, 1)$ $5 = (2, 1)$ $9 = (0, 1)$
$2 = (2, 2)$ $6 = (0, 2)$ $10 = (1, 2)$
$3 = (0, 3)$ $7 = (1, 3)$ $11 = (2, 3)$. ◀

Para realizar a aritmética com números inteiros grandes, selecionamos os módulos $m_1, m_2, \ldots, m_n$, em que cada m_i é um número inteiro maior que 2, $\text{mdc}(m_i, m_j) = 1$ sempre que $i \neq j$ e $m = m_1 m_2 \cdots m_n$ é maior que o resultado das operações aritméticas que queremos realizar.

Uma vez selecionados nossos módulos, fazemos as operações aritméticas com números inteiros grandes a partir da realização de operações nas n-uplas, representando esses números inteiros usando seus restos da divisão por m_i, $i = 1, 2, \ldots, n$. Uma vez computado o valor de cada componente no resultado, recuperamos seu valor resolvendo um sistema de n congruências módulo m_i, $i = 1, 2, \ldots, n$. Este método para realizar as operações aritméticas com números inteiros grandes tem muitas características importantes. Primeiro, pode ser usado para fazer operações aritméticas com números inteiros maiores que os números que um computador suporta. Segundo, as computações relacionadas a módulos diferentes podem ser feitas em paralelo, acelerando a aritmética.

EXEMPLO 8 Suponha que fazer a aritmética com números inteiros menores que 100 em um determinado processador seja mais rápido que fazer a aritmética com números inteiros maiores. Podemos restringir quase todas as computações a números inteiros menores que 100 se representarmos os números inteiros usando restos módulo pares de primos entre si inteiros menores que 100. Por exemplo, podemos usar os módulos 99, 98, 97 e 95. (Estes números inteiros são pares de primos entre si, pois nenhum deles tem fator comum maior que 1.)

Pelo Teorema Chinês do Resto, todo número inteiro não negativo menor que $99 \cdot 98 \cdot 97 \cdot 95 = 89\,403\,930$ pode ser representado unicamente pelos seus restos quando dividido por esses quatro módulos. Por exemplo, representamos 123 684 como (33, 8, 9, 89), pois 123 684 **mod** 99 = 33 123 684 **mod** 98 = 8; 123 684 **mod** 97 = 9 e 123 684 **mod** 95 = 89. Da mesma forma, representamos 413 456 como (32, 92, 42, 16).

Para encontrar a soma de 123 684 e 413 456, trabalhamos com esses 4-uplas em vez desses dois números inteiros diretamente. Adicionamos as 4-uplas e reduzimos cada componente com seu módulo apropriado. Isto nos dá

$$\begin{aligned}(33, 8, 9, 89) + (32, 92, 42, 16) &= (65 \textbf{ mod } 99, 100 \textbf{ mod } 98, 51 \textbf{ mod } 97, 105 \textbf{ mod } 95)\\ &= (65, 2, 51, 10).\end{aligned}$$

Para encontrar a soma, ou seja, o número inteiro representado por (65, 2, 51, 10), precisamos resolver o sistema de congruências

$$\begin{aligned}x &\equiv 65 \pmod{99}\\ x &\equiv 2 \pmod{98}\\ x &\equiv 51 \pmod{97}\\ x &\equiv 10 \pmod{95}\end{aligned}$$

Podemos mostrar (veja o Exercício 39) que 537 140 é a única solução não negativa deste sistema menor que 89 403 930. Conseqüentemente, 537 140 é a soma. Note que é apenas quando temos que recuperar o número inteiro representado por (65, 2, 51, 10) que precisamos fazer a aritmética com os números inteiros maiores que 100. ◀

Boas escolhas para os módulos na aritmética com números inteiros grandes ocorrem particularmente em um conjunto de inteiros na forma $2^k - 1$, em que k é um número inteiro positivo, porque fica fácil fazer aritmética modular binária com tais inteiros e encontrar conjuntos com tais inteiros que sejam pares de primos entre si. [A segunda razão é uma conseqüência do fato de que $\text{mdc}(2^a - 1, 2^b - 1) = 2^{\text{mdc}(a, b)} - 1$, como mostra o Exercício 41.] Suponha, por exemplo, que possamos fazer facilmente aritmética com números inteiros menores que 2^{35} em nossos computadores, mas o trabalho com números inteiros grandes necessita de procedimentos especiais. Podemos usar os módulos de primos entre si menores que 2^{35} para realizar a aritmética com números inteiros tão grandes como seu produto. Por exemplo, no Exercício 42, os números inteiros $2^{35} - 1$, $2^{34} - 1$, $2^{33} - 1$, $2^{31} - 1$, $2^{29} - 1$ e $2^{23} - 1$ são primos entre si. Como o produto desses seis módulos excede 2^{184}, podemos realizar a aritmética com números inteiros tão grandes quanto 2^{184} (contanto que os resultados não excedam esse valor) fazendo a aritmética modular para cada um desses seis módulos, nenhum excedente a 2^{35}.

Pseudoprimos

Na Seção 3.5, mostramos que um número inteiro n é primo quando não for divisível por qualquer número primo p com $p \leq \sqrt{n}$. Infelizmente, é ineficiente usar este critério para mostrar que um dado número inteiro é primo. Com isso, teríamos de encontrar todos os primos não excedentes a $\sqrt{n}$ e também realizar a divisão para cada primo para ver se ele divide n.

Existem maneiras mais eficientes de determinar se um número inteiro é primo? De acordo com algumas fontes, os matemáticos chineses acreditavam que n era primo se e somente se

$$2^{n-1} \equiv 1 \pmod{n}.$$

Se isso fosse verdade, poderia fornecer um teste mais eficiente. Por que eles acreditavam que essa congruência poderia ser usada para determinar se um número inteiro é primo? Primeiro, eles observaram que a congruência é mantida se n for primo. Por exemplo, 5 é primo e

$$2^{5-1} = 2^4 = 16 \equiv 1 \pmod{5}.$$

Em segundo lugar, eles nunca encontraram um número inteiro composto n para o qual a congruência se mantinha. Os chineses antigos estavam parcialmente corretos. Estavam certos em pensar que a congruência era mantida se n fosse primo, mas estavam equivocados em concluir que n é necessariamente primo se a congruência for mantida.

O grande matemático francês Fermat mostrou que a congruência se mantém quando n é primo. Ele demonstrou o resultado geral apresentado abaixo.

TEOREMA 5 **PEQUENO TEOREMA DE FERMAT** Se p for primo e a for um número inteiro não divisível por p, então

$$a^{p-1} \equiv 1 \pmod{p}.$$

Além disso, para todo número inteiro a, temos

$$a^p \equiv a \pmod{p}.$$

A demonstração do Teorema 5 está proposta no Exercício 17, no final desta seção.

Infelizmente, há números inteiros compostos n, tal que $2^{n-1} \equiv 1 \pmod{n}$. Tais números inteiros são chamados de **pseudoprimos** de base 2.

EXEMPLO 9 O número inteiro 341 é um pseudoprimo de base 2 porque é composto ($341 = 11 \cdot 31$) e, como mostra o Exercício 27,

$$2^{340} \equiv 1 \pmod{341}.$$ ◀

Podemos usar um número inteiro diferente de 2 como base quando estudamos pseudoprimos.

Links

PIERRE DE FERMAT (1601–1665) Pierre de Fermat, um dos mais importantes matemáticos do século XVII, era advogado. Ele é o mais famoso matemático amador da história. Fermat pouco publicou sobre suas descobertas matemáticas. É por meio de sua correspondência com outros matemáticos que conhecemos seus trabalhos. Fermat foi um dos inventores da geometria analítica e desenvolveu algumas idéias fundamentais do cálculo. Juntamente com Pascal, deu à probabilidade uma base matemática. Além disso, formulou o que foi o mais famoso problema não resolvido em matemática. Ele afirmou que a equação $x^n + y^n = z^n$ não tem soluções inteiras positivas não-triviais quando n é um número inteiro maior que 2. Por mais de 300 anos, nenhuma demonstração (ou contra-exemplo) foi encontrada. Em sua cópia dos trabalhos do matemático grego Diofanto, Fermat escreveu que ele tinha uma demonstração, mas não caberia na margem. Como a primeira demonstração, encontrada por Andrew Wiles em 1994, baseava-se na matemática moderna e sofisticada, a maioria das pessoas acha que Fermat acreditava ter uma demonstração, mas que ela era incorreta. Entretanto, ele pode ter tido outras abordagens na sua procura por uma demonstração, não sendo capaz de encontrar uma por si próprio.

DEFINIÇÃO 1 Considere b como um número inteiro positivo. Se n for um número inteiro positivo composto e $b^{n-1} \equiv 1 \pmod{n}$, então n é chamado de *pseudoprimo na base b*.

Dado um número inteiro positivo n, determinar se $2^{n-1} \equiv 1 \pmod{n}$ é um teste útil que fornece alguma evidência sobre se n é primo. Em particular, se n satisfizer essa congruência, então ele é primo ou pseudoprimo na base 2; se n não satisfizer essa congruência, ele é composto. Podemos realizar testes semelhantes usando a base b diferente de 2 e obter mais evidências de que n é primo. Se n passar em todos esses testes, ele é primo ou pseudoprimo em todas as bases b que escolhermos. Além disso, entre os números inteiros positivos não excedentes a x, em que x é um número real positivo, comparado aos primos, existem relativamente poucos pseudoprimos na base b, em que b é um número inteiro positivo. Por exemplo, menores que 10^{10}, existem 455 052 512 primos, mas apenas 14 884 pseudoprimos na base 2. Infelizmente, não podemos distinguir entre primos e pseudoprimos apenas escolhendo bases suficientes, porque há números inteiros compostos n que passam em todos os testes em bases com $\text{mdc}(b, n) = 1$. Isto nos leva à Definição 2.

DEFINIÇÃO 2 Um número inteiro composto n que satisfaça a congruência $b^{n-1} \equiv 1 \pmod{n}$ para todos os números inteiros positivos b com $\text{mdc}(b, n) = 1$ é chamado de *número de Carmichael*. (O nome desses números remete a Robert Carmichael, que os estudou no começo do século XX.)

EXEMPLO 10 O número inteiro 561 é um número de Carmichael. Para verificar isso, primeiro note que 561 é composto, porque $561 = 3 \cdot 11 \cdot 17$. Depois, note que se $\text{mdc}(b, 561) = 1$, então $\text{mdc}(b, 3) = \text{mdc}(b, 11) = \text{MDC}(b, 17) = 1$.

Usando o Pequeno Teorema de Fermat, vemos que

$$b^2 \equiv 1 \pmod{3},\ b^{10} \equiv 1 \pmod{11} \text{ e } b^{16} \equiv 1 \pmod{17}.$$

Portanto,

$$b^{560} = (b^2)^{280} \equiv 1 \pmod{3},$$
$$b^{560} = (b^{10})^{56} \equiv 1 \pmod{11},$$
$$b^{560} = (b^{16})^{35} \equiv 1 \pmod{17}.$$

No Exercício 23 no final desta seção, temos que $b^{560} \equiv 1 \pmod{561}$ para todo número inteiro positivo b com $\text{mdc}(b, 561) = 1$. Assim, 561 é um número de Carmichael. ◀

Embora haja muitos números de Carmichael, testes mais adequados, descritos nos exercícios no final desta seção, podem ser feitos para ser usados como base para testes de proba-

---

ROBERT DANIEL CARMICHAEL (1879–1967) Robert Daniel Carmichael nasceu no Alabama. Recebeu seu diploma na Lineville College em 1898 e seu Ph.D. em 1911, em Princeton. Carmichael teve alguns cargos na Universidade da Indiana de 1911 a 1915 e na Universidade de Illinois de 1915 a 1947. Ele era um pesquisador ativo em diversas áreas, incluindo teoria dos números, análise real, equações diferenciais, física matemática e teoria dos grupos. Sua tese de Ph.D., escrita sob a orientação de G. D. Birkhoff, é considerada a primeira contribuição norte-americana significativa sobre o assunto equações diferenciais.

bilidade de primalidade, cuja verificação é mais eficiente, a fim de verificar se um número inteiro é primo. Esses testes podem ser usados para mostrar rapidamente a quase certeza de que um número inteiro dado é primo. Mais precisamente, se um número inteiro não for primo, então a probabilidade de que ele passe na série de testes é próxima de 0. Descreveremos tal teste no Capítulo 6 e discutiremos as noções de teoria da probabilidade que ele pede. Esses testes podem ser usados, e o são, para encontrar números primos grandes de maneira extremamente rápida em computadores.

Criptografia de Chave Pública

Na Seção 3.4, introduzimos métodos para codificar mensagens com base em congruências. Quando esses métodos de codificação são usados, as mensagens, que são seqüências de caracteres, são transcritas em números. Então, o número para cada caractere é transformado em outro número, se usarmos uma substituição ou uma transformação afim módulo 26. Esses métodos são exemplos de **criptossistemas de chave privada.** Conhecer a chave de codificação permite que encontremos rapidamente a chave de decodificação. Por exemplo, quando um código de substituição é usado com chave de codificação k, um número p representando uma letra é enviado a

$$c = (p + k) \bmod 26.$$

A decodificação é realizada pela substituição de $-k$; ou seja,

$$p = (c - k) \bmod 26.$$

Quando um criptossistema de chave privada é usado, um par de pessoas que desejam se comunicar em segredo precisa ter uma chave separada. Como qualquer um que conheça essa chave pode facilmente codificar e decodificar as mensagens, essas duas pessoas precisam trocar a chave para segurança.

Em meados da década de 70, criptologistas introduziram o conceito de **criptossistemas de chave pública.** Quando tais criptossistemas são usados, saber como enviar a alguém uma mensagem não ajuda na decodificação de mensagens enviadas a essa pessoa. Em tal sistema, toda pessoa pode ter uma chave pública. Apenas as chaves de decodificação são mantidas em segredo e apenas o destinatário da mensagem pode decodificá-la, porque a chave de codificação não permite que alguém encontre a chave de decodificação sem ter um trabalho muito grande (algo como 2 bilhões de anos em tempo de computação).

O Criptossistema RSA

Em 1976, três pesquisadores do M.I.T. — Ronald Rivest, Adi Shamir e Leonard Adleman — apresentaram ao mundo um criptossistema de chave pública, conhecido como **sistema RSA,** a partir das iniciais de seus inventores. (Como geralmente acontece com as descobertas criptográficas, o sistema RSA já tinha sido descoberto em uma pesquisa secreta do governo. Clifford Cocks, trabalhando secretamente para o governo do Reino Unido, descobriu este criptossistema em 1973, mas sua invenção ficou desconhecida até o final da década de 90. Um excelente material sobre essa antiga descoberta, assim como o trabalho de Rivest, Shamir e Adleman, pode ser encontrado em [Si99].) O criptossistema RSA é baseado em potenciação modular do produto de dois números primos grandes, o que pode ser feito rapidamente usando o Algoritmo 5 da Seção 3.6. Cada indivíduo tem uma chave de codificação que consiste em um módulo $n = pq$, em que p e q são números primos grandes, com 200 dígitos cada, e um expoente e que é relativamente primo a $(p - 1)(q - 1)$. Para produzir uma chave que possa ser usada na decodificação, dois números primos grandes devem ser encontrados. Isto pode ser feito rapidamente em um computador usando-se os testes de probabilidade de primalidade, mencionados anteriormente nesta seção. Entretanto, o produto desses números primos $n = pq$, com aproximadamente 400 dígitos, não pode ser fatorado em uma quantidade razoável de tempo. Como veremos, esta é uma razão importante do porquê a decodificação não pode ser feita rapidamente sem uma chave de decodificação separada.

Codificação RSA

No método de codificação RSA, as mensagens são transcritas em seqüências de números inteiros. Isto pode ser feito transcrevendo cada letra em um número inteiro, como foi feito com o código de César. Esses números inteiros são agrupados na forma de inteiros grandes, cada um representando um bloco de letras. A codificação é feita transformando o número inteiro M, que representa o texto total (a mensagem original), em um número inteiro C, que representa o texto codificado (a mensagem codificada), usando-se a função

$$C = M^e \bmod n.$$

(Para realizar a codificação, usamos um algoritmo para potenciação modular rápida, assim como o Algoritmo 5 da Seção 3.6.) Deixamos a mensagem codificada como blocos de números e os enviamos ao devido destinatário.

O Exemplo 11 ilustra como a codificação RSA é realizada. Por motivos de ordem prática, usamos números primos pequenos p e q no exemplo, em vez de primos com 100 dígitos ou mais. Embora o código descrito neste exemplo não seja seguro, ele ilustra as técnicas usadas no código RSA.

EXEMPLO 11 Codifique a mensagem STOP usando o criptossistema RSA com $p = 43$ e $q = 59$, para que $n = 43 \cdot 59 = 2537$, com $e = 13$. Note que

$$\text{mdc}(e, (p-1)(q-1)) = \text{mdc}(13, 42 \cdot 58) = 1.$$

Solução: Transcrevemos as letras de STOP em equivalentes numéricos e então agrupamos os números em blocos de quatro. Obtemos

Links

1819 1415.

RONALD RIVEST (Nascido em 1948) Ronald Rivest recebeu seu doutorado em Yale, em 1969, e seu Ph.D. em ciência da computação em Stanford, em 1974. Ele é professor de ciência da computação no M.I.T. e foi um dos fundadores da Segurança de Dados RSA, que mantém a patente do criptossistema RSA que ele inventou juntamente com Adi Shamir e Leonard Adleman. As áreas em que Rivest tem trabalhado, além da criptografia, incluem inteligência artificial, design de VLSI e algoritmos. Ele é co-autor de um texto popular sobre algoritmos ([CoLeRiSt01]).

ADI SHAMIR (Nascido em 1952) Adi Shamir nasceu em Tel Aviv, em Israel. Ele obteve seu título de bacharel pela Universidade de Tel Aviv (1972) e seu Ph.D. pelo Weizmann Institute of Science (1977). Shamir foi pesquisador-assistente na Universidade de Warwick e professor-assistente no M.I.T. Atualmente, é professor do Departamento de Matemática Aplicada do Weizmann Institute e lidera um grupo de estudo sobre segurança computacional. As contribuições de Shamir à criptografia, além do criptossistema RSA, incluem criptossistemas de craqueamento de pacotes, análise criptográfica do Data Encryption Standard (DES) e a criação de muitos protocolos de criptografia.

LEONARD ADLEMAN (Nascido em 1945) Leonard Adleman nasceu em São Francisco, Califórnia. Recebeu seu título de bacharel em matemática (1968) e seu Ph.D. em ciência da computação (1976) pela Universidade da Califórnia, em Berkeley. Adleman foi membro da faculdade de matemática do M.I.T. de 1976 a 1980, onde foi co-inventor do criptossistema RSA, e em 1980 conquistou um cargo no departamento de ciência da computação na University of Southern California (USC). Ele foi indicado a uma cadeira na USC em 1985. Adleman trabalha com segurança computacional, complexidade computacional, imunologia e biologia molecular. Ele inventou o termo "vírus de computador". O trabalho recente de Adleman em computar o DNA tem despertado grande interesse. Ele foi consultor técnico do filme *Quebra de sigilo*, no qual a segurança computacional desenvolve um importante papel.

Codificamos cada bloco usando a função

$$C = M^{13} \bmod 2537.$$

Computações que usam multiplicações modulares mostram que $1819^{13} \bmod 2537 = 2081$ e $1415^{13} \bmod 2537 = 2182$. A mensagem codificada é 2081 2182. ◀

Decodificação RSA

A mensagem de texto puro pode ser rapidamente recuperada quando a chave de decodificação d, um inverso de e módulo $(p-1)(q-1)$, é conhecida. [Tal que um inverso exista, já que $\text{mdc}(e, (p-1)(q-1)) = 1$.] Para verificar isso, note que se $de \equiv 1 \pmod{(p-1)(q-1)}$, há um número inteiro k, tal que $de = 1 + k(p-1)(q-1)$. Portanto,

$$C^d \equiv (M^e)^d = M^{de} = M^{1+k(p-1)(q-1)} \pmod{n}.$$

Pelo Pequeno Teorema de Fermat [supondo que $\text{mdc}(M, p) = \text{mdc}(M, q) = 1$, que é mantido, com exceção de casos raros], temos que $M^{p-1} \equiv 1 \pmod{p}$ e $M^{q-1} \equiv 1 \pmod{q}$. Conseqüentemente,

$$C^d \equiv M \cdot (M^{p-1})^{k(q-1)} \equiv M \cdot 1 \equiv M \pmod{p}$$

e

$$C^d \equiv M \cdot (M^{q-1})^{k(p-1)} \equiv M \cdot 1 \equiv M \pmod{q}.$$

Como $\text{mdc}(p, q) = 1$, temos que, pelo Teorema Chinês do Resto,

$$C^d \equiv M \pmod{pq}.$$

O Exemplo 12 ilustra como decodificar mensagens enviadas usando o criptossistema RSA.

EXEMPLO 12 Recebemos a mensagem codificada 0981 0461. Qual é a mensagem decodificada se ela tiver sido codificada usando-se o código RSA do Exemplo 11?

Solução: A mensagem foi codificada usando-se o criptossistema RSA com $n = 43 \cdot 59$ e expoente 13. Como mostra o Exercício 4, $d = 937$ é um inverso de 13 módulo $42 \cdot 58 = 2436$. Usamos 937 como nosso expoente de decodificação. Conseqüentemente, para decodificar um bloco C, computamos

$$P = C^{937} \bmod 2537.$$

Para decodificar a mensagem, usamos um algoritmo de potenciação modular rápido para computar $0981^{937} \bmod 2537 = 0704$ e $0461^{937} \bmod 2537 = 1115$. Conseqüentemente, a versão numérica da mensagem original é 0704 1115. Traduzindo de volta às letras, vemos que a mensagem é HELP. ◀

RSA como um Sistema de Chave Pública

Links Por que o criptossistema RSA é adequado para a criptografia de chave pública? Quando sabemos a fatoração do módulo n, ou seja, quando conhecemos p e q, podemos usar o algoritmo de Euclides para encontrar rapidamente um expoente d inverso a e módulo $(p-1)(q-1)$. Isto nos permite decodificar mensagens enviadas usando nossa chave. Entretanto, não há método conhecido para decodificar mensagens que não seja baseado em encontrar uma fatoração de n, ou então fugir

da fatoração de n. Acredita-se que a fatoração seja um problema difícil, oposto a encontrar os primos grandes p e q, o que pode ser feito rapidamente. O método de fatoração mais eficiente conhecido (como de 2005) requer bilhões de anos para fatorar números inteiros com 400 dígitos. Conseqüentemente, quando p e q são números primos com 200 dígitos, as mensagens codificadas usando-se $n = pq$ como módulo não podem ser encontradas em um tempo razoável, a menos que os números primos p e q sejam conhecidos.

A pesquisa ativa está a caminho de encontrar novas maneiras para tornar mais eficiente a fatoração de números inteiros. Aqueles que eram considerados, recentemente e muitos anos atrás, muito grandes para serem fatorados em um tempo razoável, agora podem ser fatorados rapidamente. Números inteiros com mais de 100 dígitos, assim como alguns com mais de 150 dígitos, têm sido fatorados em um trabalho em equipe. Quando novas técnicas de fatoração forem descobertas, será necessário usar números primos maiores para assegurar o segredo das mensagens. Infelizmente, as mensagens que foram consideradas seguras anteriormente podem ser salvas e decodificadas por vários destinatários quando o fator de $n = pq$ torna-se acessível na chave usada pela codificação RSA.

O método RSA é agora mundialmente utilizado. Entretanto, os criptossistemas mais usados são os de chave privada. O uso da criptografia de chave pública, via sistema RSA, está crescendo. Entretanto, há aplicações que usam sistemas de chaves privadas e públicas. Por exemplo, um criptossistema de chave pública, como o RSA, pode ser usado para distribuir chaves privadas a pares de indivíduos quando eles desejam se comunicar. Essas pessoas usam, então, um sistema de chave privada para codificar e decodificar mensagens.

Exercícios

1. Expresse o máximo divisor comum de cada par de números inteiros como uma combinação linear destes inteiros.
a) 10, 11 **b)** 21, 44 **c)** 36, 48
d) 34, 55 **e)** 117, 213 **f)** 0, 223
g) 123, 2347 **h)** 3454, 4666 **i)** 9999, 11 111

2. Expresse o máximo divisor comum de cada par de números inteiros como uma combinação linear destes inteiros.
a) 9, 11 **b)** 33, 44 **c)** 35, 78
d) 21, 55 **e)** 101, 203 **f)** 124, 323
g) 2002, 2339 **h)** 3457, 4669 **i)** 10 001, 13 422

3. Mostre que 15 é um inverso de 7 módulo 26.

4. Mostre que 937 é um inverso de 13 módulo 2436.

5. Encontre um inverso de 4 módulo 9.

6. Encontre um inverso de 2 módulo 17.

7. Encontre um inverso de 19 módulo 141.

8. Encontre um inverso de 144 módulo 233.

***9.** Mostre que se a e m forem inteiros positivos primos entre si, então o inverso de a módulo m é o único módulo m. [*Dica:* Suponha que haja duas soluções b e c da congruência de $ax \equiv 1 \pmod m$. Use o Teorema 2 para mostrar que $b \equiv c \pmod m$.]

10. Mostre que um inverso de a módulo m não existe se $\text{mdc}(a, m) > 1$.

11. Resolva a congruência $4x \equiv 5 \pmod 9$.

12. Resolva a congruência $2x \equiv 7 \pmod{17}$.

***13.** Mostre que se m for um número inteiro positivo maior que 1 e $ac \equiv bc \pmod m$, então $a \equiv b \bmod m/\text{mdc}(c, m)$.

14. **a)** Mostre que os números inteiros positivos menores que 11, exceto 1 e 10, podem ser separados em pares de inteiros, de tal modo que cada par consista de inteiros que são inversos de outro módulo 11.
b) Use o item (a) para mostrar que $10! \equiv -1 \pmod{11}$.

15. Mostre que se p é primo, as únicas soluções de $x^2 \equiv 1 \pmod p$ são números inteiros x, tal que $x \equiv 1 \pmod p$ e $x \equiv -1 \pmod p$.

***16.** **a)** Generalize o resultado do item (a) do Exercício 14; ou seja, mostre que se p for um número primo, os números inteiros positivos menores que p, exceto 1 e $p - 1$, podem ser divididos em $(p - 3)/2$ pares de inteiros, tal que cada par tenha inteiros que sejam inversos uns dos outros. [*Dica:* Use o resultado do Exercício 15.]
b) A partir do item (a), conclua que $(p - 1)! \equiv -1 \pmod p$ sempre que p for primo. Este resultado é conhecido como **Teorema de Wilson.**
c) O que podemos concluir se n for um número inteiro positivo, tal que $(n - 1)! \not\equiv -1 \pmod n$?

***17.** Este exercício refere-se à demonstração do Pequeno Teorema de Fermat.
a) Suponha que a não seja divisível pelo primo p. Mostre que os inteiros $1 \cdot a, 2 \cdot a, \ldots, (p - 1)a$ não são congruentes módulo p dois a dois.
b) Conclua, a partir do item (a), que o produto de $1, 2, \ldots, p - 1$ é congruente módulo p ao produto de $a, 2a, \ldots, (p - 1)a$. Use esta informação para mostrar que
$$(p - 1)! \equiv a^{p-1}(p - 1)! \pmod p.$$
c) Use o Teorema 2 para mostrar, a partir do item (b), que $a^{p-1} \equiv 1 \pmod p$ se $p \not| a$. [*Dica:* Use o Lema 2 para mostrar que p não divide $(p - 1)!$ e então use o Teorema 2. Outra maneira é usar o Teorema de Wilson.]

d) Use o item (c) para mostrar que $a^p \equiv a \pmod{p}$ para todos os números inteiros a.

18. Encontre todas as soluções para o sistema de congruências.
$x \equiv 2 \pmod 3$
$x \equiv 1 \pmod 4$
$x \equiv 3 \pmod 5$

19. Encontre todas as soluções para o sistema de congruências.
$x \equiv 1 \pmod 2$
$x \equiv 2 \pmod 3$
$x \equiv 3 \pmod 5$
$x \equiv 4 \pmod{11}$

***20.** Encontre todas as soluções, se houver, para o sistema de congruências.
$x \equiv 5 \pmod 6$
$x \equiv 3 \pmod{10}$
$x \equiv 8 \pmod{15}$

***21.** Encontre todas as soluções, se houver, para o sistema de congruências.
$x \equiv 7 \pmod 9$
$x \equiv 4 \pmod{12}$
$x \equiv 16 \pmod{21}$

22. Use o Teorema Chinês do Resto para mostrar que um número inteiro a, com $0 \leq a < m = m_1 m_2 \cdots m_n$, em que os números inteiros $m_1, m_2, \ldots, m_n$ são pares de primos entre si, pode ser representado unicamente pela n-upla (a **mod** m_1, a **mod** m_2, . . . , a **mod** m_n).

***23.** Considere $m_1, m_2, \ldots, m_n$ pares de primos entre si maiores que ou iguais a 2. Mostre que se $a \equiv b \pmod{m_i}$ para $i = 1, 2, \ldots, n$, então $a \equiv b \pmod m$, em que $m = m_1 m_2 \cdots m_n$. (Este resultado será usado no Exercício 24 para demonstrar o Teorema Chinês do Resto. Conseqüentemente, não use esse teorema para demonstrá-lo.)

***24.** Complete a demonstração do Teorema Chinês do Resto mostrando que a solução simultânea de um sistema de congruências lineares módulo primos entre si dois a dois é única módulo produto destes módulos. [*Dica:* Suponha que x e y são duas soluções simultâneas. Mostre que $m_i \mid x - y$ para todo i. Usando o Exercício 23, conclua que $m = m_1 m_2 \cdots m_n \mid x - y$.]

25. Quais números inteiros tem resto 1 quando dividido por 2 e resto também 1 quando dividido por 3?

26 Quais números inteiros são divisíveis por 5, mas têm resto 1 quando divididos por 3?

27. a) Mostre que $2^{340} \equiv 1 \pmod{11}$ pelo Pequeno Teorema de Fermat e note que $2^{340} = (2^{10})^{34}$.

b) Mostre que $2^{340} \equiv 1 \pmod{31}$ usando o fato de que $2^{340} = (2^5)^{68} = 32^{68}$.

c) Conclua, a partir dos itens (a) e (b), que $2^{340} \equiv 1 \pmod{341}$.

28. a) Use o Pequeno Teorema de Fermat para computar 3^{302} **mod** 5, 3^{302} **mod** 7 e 3^{302} **mod** 11.

b) Use seus resultados do item (a) e o Teorema Chinês do Resto para encontrar 3^{302} **mod** 385. (Note que $385 = 5 \cdot 7 \cdot 11$.)

29. a) Use o Pequeno Teorema de Fermat para computar 5^{2003} **mod** 7, 5^{2003} **mod** 11 e 5^{2003} **mod** 13.

b) Use seus resultados do item (a) e o Teorema Chinês do Resto para encontrar 5^{2003} **mod** 1001. (Note que $1001 = 7 \cdot 11 \cdot 13$.)

☞Considere n como um número inteiro positivo e $n - 1 = 2^s t$, em que s é um número inteiro não negativo e t é um número inteiro positivo ímpar. Dizemos que n passa no **teste de Miller para a base** b se $b^t \equiv 1 \pmod n$ ou $b^{2^j t} \equiv -1 \pmod n$ para qualquer j com $0 \leq j \leq s - 1$. Podemos mostrar (veja [Ro05]) que um número inteiro composto n passa no teste de Miller para menos que $n/4$ bases b com $1 < b < n$.

***30.** Mostre que se n é um número primo e b é um número inteiro positivo com $n \nmid b$, então n passa no teste de Miller para a base b.

31. Mostre que 2047 passa no teste de Miller para base 2, mas que ele é composto. Um número inteiro composto positivo n que passa no teste de Miller para base b é chamado de **pseudoprimo forte de base** b. Temos que 2047 é um pseudoprimo forte de base 2.

32. Mostre que 1729 é um número de Carmichael.

33. Mostre que 2821 é um número de Carmichael.

***34.** Mostre que se $n = p_1 p_2 \cdots p_k$, em que $p_1, p_2, \ldots, p_k$ são números primos distintos que satisfazem $p_j - 1 \mid n - 1$ para $j = 1, 2, \ldots, k$, então n é um número de Carmichael.

35. a) Use o Exercício 34 para mostrar que todo número inteiro na forma $(6m + 1)(12m + 1)(18m + 1)$, em que m é um número inteiro positivo e $6m + 1$, $12m + 1$ e $18m + 1$ são todos primos, é um número de Carmichael.

b) Use o item (a) para mostrar que 172 947 529 é um número de Carmichael.

36. Encontre o número inteiro não negativo a menor que 28 representado por estes pares, em que cada par representa (a **mod** 4, a **mod** 7).

a) (0, 0)	**b)** (1, 0)	**c)** (1, 1)
d) (2, 1)	**e)** (2, 2)	**f)** (0, 3)
g) (2, 0)	**h)** (3, 5)	**i)** (3, 6)

37. Expresse cada número inteiro não negativo a menor que 15 usando o par (a **mod** 3, a **mod** 5).

38. Explique como usar os pares encontrados no Exercício 37 para somar 4 e 7.

39. Resolva o sistema de congruências que aparece no Exemplo 8.

***40.** Mostre que sempre que a e b forem números inteiros positivos, então $(2^a - 1)$ **mod** $(2^b - 1) = 2^{a \bmod b} - 1$.

****41.** Use o Exercício 40 para mostrar que se a e b forem números inteiros positivos, então $\text{mdc}(2^a - 1, 2^b - 1) = 2^{\text{mdc}(a, b)} - 1$. [*Dica:* Mostre que os restos obtidos quando o algoritmo de Euclides é usado para computar $\text{mdc}(2^a - 1, 2^b - 1)$ estão na forma $2^r - 1$, em que r é um resto que aparece quando o algoritmo de Euclides é usado para encontrar $\text{mdc}(a, b)$.]

42. Use o Exercício 41 para mostrar que os números inteiros $2^{35} - 1$, $2^{34} - 1$, $2^{33} - 1$, $2^{31} - 1$, $2^{29} - 1$ e $2^{23} - 1$ são pares de primos entre si.

43. Mostre que se p for um primo ímpar, então todo divisor do número de Mersenne $2^p - 1$ está na forma $2kp + 1$, em que

k é um número inteiro não negativo. [*Dica:* Use o Pequeno Teorema de Fermat e o Exercício 41.]

44. Use o Exercício 43 para determinar se $M_{13} = 2^{13} - 1 = 8191$ e $M_{23} = 2^{23} - 1 = 8\,388\,607$ são números primos.

***45.** Mostre que podemos fatorar facilmente n quando sabemos que n é o produto de dois primos, p e q, e quando sabemos o valor de $(p - 1)(q - 1)$.

46. Codifique a mensagem ATTACK usando o sistema RSA, com $n = 43 \cdot 59$ e $e = 13$, transcrevendo cada letra em números inteiros e agrupando pares de inteiros, como foi feito no Exemplo 11.

47. Qual é a mensagem original codificada usando-se o sistema RSA, com $n = 43 \cdot 59$ e $e = 13$, se a mensagem codificada for 0667 1947 0671? (*Nota:* Algum auxílio computacional é necessário para fazer isso em tempo real.)

O **algoritmo de Euclides estendido** pode ser usado para expressar mdc(a, b) como uma combinação linear com coeficientes inteiros dos números inteiros a e b. Verificamos $s_0 = 1$, $s_1 = 0$, $t_0 = 0$ e $t_1 = 1$ e consideramos $s_j = s_{j-2} - q_{j-1}s_{j-1}$ e $t_j = t_{j-2} - q_{j-1}t_{j-1}$ para $j = 2, 3, \ldots, n$, em que q_j são os quocientes nas divisões usadas quando o algoritmo de Euclides encontra mdc(a, b) (veja a página 228). Podemos mostrar (veja [Ro05]) que mdc(a, b) $= s_n a + t_n b$.

48. Use o algoritmo de Euclides estendido para expressar mdc(252, 356) como uma combinação linear de 252 e 356.

49. Use o algoritmo de Euclides estendido para expressar mdc(144, 89) como uma combinação linear de 144 e 89.

50. Use o algoritmo de Euclides estendido para expressar mdc(1001, 100001) como uma combinação linear de 1001 e 100001.

51. Descreva o algoritmo de Euclides estendido usando o pseudocódigo.

Se m for um número inteiro positivo, o inteiro a é um **resíduo quadrático** de m se mdc(a, m) $= 1$ e a congruência $x^2 \equiv a \pmod{m}$ tem uma solução. Em outras palavras, um resíduo quadrático de m é um número inteiro relativamente primo de m que é um quadrado perfeito módulo m. Por exemplo, 2 é um resíduo quadrático de 7 porque mdc(2, 7) $= 1$ e $3^2 \equiv 2 \pmod 7$, e 3 é um resíduo não quadrático de 7 porque mdc(3, 7) $= 1$ e $x^2 \equiv 3 \pmod 7$ não tem solução.

52. Quais números inteiros são resíduos quadráticos de 11?

53. Mostre que se p for um número primo ímpar e a for um número inteiro não divisível por p, então a congruência $x^2 \equiv a \pmod p$ não tem solução ou tem duas soluções não congruentes módulo p.

54. Mostre que se p é um número primo ímpar, então existem $(p - 1)/2$ resíduos quadráticos de p entre os inteiros 1, 2, . . . , $p - 1$.

Se p for um número primo ímpar e a for um número inteiro não divisível por p, o **símbolo de Legendre** $\left(\frac{a}{p}\right)$ é definido por 1 se a for um resíduo quadrático de p e -1 caso contrário.

55. Mostre que se p é um número primo ímpar e a e b são números inteiros com $a \equiv b \pmod p$, então

$$\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right).$$

56. Demonstre que se p for um número primo ímpar e a for um número inteiro positivo não divisível por p, então

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod p.$$

57. Use o Exercício 56 para mostrar que se p for um número primo ímpar e a e b forem números inteiros não divisíveis por p, então

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right).$$

58. Mostre que se p for um número primo ímpar, então -1 é um resíduo quadrático de p se $p \equiv 1 \pmod 4$ e -1 não é um resíduo quadrático de p se $p \equiv 3 \pmod 4$. [*Dica:* Use o Exercício 56.]

59. Encontre todas as soluções da congruência $x^2 \equiv 29 \pmod{35}$. [*Dica:* Encontre as soluções dessa congruência módulo 5 e módulo 7, e então use o Teorema do Resto Chinês.]

60. Encontre todas as soluções da congruência $x^2 \equiv 16 \pmod{105}$. [*Dica:* Encontre as soluções dessa congruência módulo 3, módulo 5 e módulo 7, e então use o Teorema do Resto Chinês.]

***61.** Explique por que não seria adequado usar p, sendo p um primo grande, como o módulo para codificar no criptossistema RSA. Ou seja, explique como alguém poderia, sem computação excessiva, encontrar uma chave privada a partir de uma chave pública correspondente se o módulo fosse um número primo grande, em vez do produto de dois números primos grandes.

3.8 Matrizes

Introdução

As matrizes são usadas na matemática discreta para expressar relações entre elementos nos conjuntos. Nos próximos capítulos, usaremos vários modelos de matrizes. Por exemplo, elas serão usadas em modelos de redes de comunicações e sistemas de transporte. Muitos algoritmos serão desenvolvidos com esses modelos de matrizes. Esta seção revisa a aritmética de matrizes que será usada nos algoritmos.

DEFINIÇÃO 1 Uma *matriz* é uma disposição retangular de números. Uma matriz com m linhas e n colunas é chamada de matriz $m \times n$. O plural de matriz é *matrizes.* Uma matriz com o mesmo número de linhas e colunas é chamada de *quadrada.* Duas matrizes são *iguais* se elas tiverem o mesmo número de linhas, o mesmo número de colunas e se as entradas correspondentes em todas as posições forem iguais.

EXEMPLO 1 A matriz $\begin{bmatrix} 1 & 1 \\ 0 & 2 \\ 1 & 3 \end{bmatrix}$ é uma matriz 3×2. ◀

Introduziremos agora alguma terminologia sobre matrizes. As letras em maiúsculo e em negrito serão usadas para representar as matrizes.

DEFINIÇÃO 2 Considere

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}.$$

A i-ésima *linha* de $\mathbf{A}$ é a matriz $1 \times n$ $[a_{i1}, a_{i2}, \ldots, a_{in}]$. A j-ésima *coluna* de $\mathbf{A}$ é a matriz $n \times 1$

$$\begin{bmatrix} a_{1j} \\ a_{2j} \\ \cdot \\ \cdot \\ \cdot \\ a_{nj} \end{bmatrix}.$$

O (i, j)-ésimo *elemento* ou *entrada* de $\mathbf{A}$ é o elemento a_{ij}, ou seja, o número na i-ésima linha e j-ésima coluna de $\mathbf{A}$. Uma abreviação conveniente para expressar a matriz $\mathbf{A}$ é escrever $\mathbf{A} = [a_{ij}]$, que indica que $\mathbf{A}$ é a matriz com seu (i, j)-ésimo elemento igual a a_{ij}.

Aritmética de Matrizes

As operações básicas da aritmética de matrizes serão agora discutidas, começando com uma definição de adição de matrizes.

DEFINIÇÃO 3 Considere $\mathbf{A} = [a_{ij}]$ e $\mathbf{B} = [b_{ij}]$ como matrizes $m \times n$. A *soma* de $\mathbf{A}$ e $\mathbf{B}$, indicada por $\mathbf{A} + \mathbf{B}$, é a matriz $m \times n$ que tem $a_{ij} + b_{ij}$ como seu (i, j)-ésimo elemento. Em outras palavras, $\mathbf{A} + \mathbf{B} = [a_{ij} + b_{ij}]$.

A soma de duas matrizes de mesmo tamanho é obtida pela adição dos elementos nas posições correspondentes. As matrizes de tamanhos diferentes não podem ser somadas, porque a soma de duas matrizes é definida apenas quando elas tiverem o mesmo número de linhas e o mesmo número de colunas.

EXEMPLO 2 Temos $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 2 & -3 \\ 3 & 4 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 4 & -1 \\ 1 & -3 & 0 \\ -1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 & -2 \\ 3 & -1 & -3 \\ 2 & 5 & 2 \end{bmatrix}$. ◀

Discutiremos agora o produto de matrizes. Um produto de duas matrizes é definido apenas quando o número de colunas da primeira matriz é igual ao número de linhas da segunda matriz.

DEFINIÇÃO 4 Sejam **A** uma matriz $m \times k$ e **B** uma matriz $k \times n$. O *produto* de **A** e **B,** indicado por **AB,** é a matriz $m \times n$ com sua (i, j)-ésima entrada igual à soma dos produtos dos elementos correspondentes da i-ésima linha de **A** e a j-ésima coluna de **B.** Em outras palavras, se $\mathbf{AB} = [c_{ij}]$, então

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj}.$$

Na Figura 1, a linha cinza de **A** e a coluna cinza de **B** são usadas para computar o elemento c_{ij} de **AB.** O produto de duas matrizes não é definido quando o número de colunas da primeira matriz e o número de linhas da segunda matriz são diferentes.

Daremos agora alguns exemplos de produtos de matrizes.

EXEMPLO 3 Considere

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 4 \\ 2 & 1 & 1 \\ 3 & 1 & 0 \\ 0 & 2 & 2 \end{bmatrix} \text{ e } \mathbf{B} = \begin{bmatrix} 2 & 4 \\ 1 & 1 \\ 3 & 0 \end{bmatrix}.$$

Encontre **AB** se ela existir.

Exemplos Extras

Solução: Como **A** é uma matriz 4×3 e **B** é uma matriz 3×2, o produto **AB** é definido e será uma matriz 4×2. Para encontrar os elementos de **AB,** os elementos correspondentes nas linhas de **A** e nas colunas de **B** são primeiro multiplicados e então são somados. Por exemplo, o elemento na posição (3, 1) de **AB** é a soma dos produtos dos elementos correspondentes da terceira linha de **A** e da primeira coluna de **B;** ou seja, $3 \cdot 2 + 1 \cdot 1 + 0 \cdot 3 = 7$. Quando todos os elementos de

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ik} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mk} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1j} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2j} & \cdots & b_{2n} \\ \vdots & & & \vdots & & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kj} & \cdots & b_{kn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & & c_{ij} & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}$$

FIGURA 1 **O produto de $\mathbf{A} = [a_{ij}]$ e $\mathbf{B} = [b_{ij}]$.**

Quando **AB** são computados, vemos que

$$\mathbf{AB} = \begin{bmatrix} 14 & 4 \\ 8 & 9 \\ 7 & 13 \\ 8 & 2 \end{bmatrix}.$$

◀

A multiplicação de matrizes *não* é comutativa. Ou seja, se **A** e **B** são duas matrizes, não é necessariamente verdade que **AB** e **BA** sejam iguais. De fato, pode ser que apenas um desses dois produtos seja definido. Por exemplo, se **A** é 2×3 e **B** é 3×4, então **AB** é definida e será 2×4; entretanto, **BA** não é definida, pois é impossível multiplicar uma matriz 3×4 por uma matriz 2×3.

Em geral, suponha que **A** seja uma matriz $m \times n$ e **B** seja uma matriz $r \times s$. Então, **AB** é definida apenas quando $n = r$ e **BA** é definida apenas quando $s = m$. Além disso, mesmo quando **AB** e **BA** forem definidas, elas não terão o mesmo tamanho, a não ser que $m = n = r = s$. Assim, se **AB** e **BA** são definidas e são do mesmo tamanho, então **A** e **B** devem ser matrizes quadradas e de mesmo tamanho (mesma ordem). Além disso, mesmo com **A** e **B** matrizes $n \times n$, **AB** e **BA** não são necessariamente iguais, como demonstra o Exemplo 4.

EXEMPLO 4 Considere

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \text{ e } \mathbf{B} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

AB = BA?

Solução: Encontramos

$$\mathbf{AB} = \begin{bmatrix} 3 & 2 \\ 5 & 3 \end{bmatrix} \text{ e } \mathbf{BA} = \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}.$$

Assim, $\mathbf{AB} \neq \mathbf{BA}$. ◀

Algoritmos para a Multiplicação de Matrizes

A definição do produto de duas matrizes nos leva a um algoritmo que computa o produto de duas matrizes. Suponha que $\mathbf{C} = [c_{ij}]$ seja a matriz $m \times n$, que é o produto da matriz $m \times k$ $\mathbf{A} = [a_{ij}]$ pela matriz $k \times n$ $\mathbf{B} = [b_{ij}]$. O algoritmo baseado na definição do produto da matriz é expresso em pseudocódigo no Algoritmo 1.

ALGORITMO 1 Multiplicação de Matrizes.

```
procedure multiplicacao de matriz (A, B: matrizes)
for i := 1 to m
    for j := 1 to n
    begin
        c_ij := 0
        for q := 1 to k
            c_ij := c_ij + a_iq b_qj
    end
{C = [c_ij] é o produto de A e B}
```

Podemos determinar a complexidade deste algoritmo em termos do número de adições e multiplicações usadas.

EXEMPLO 5 Quantas adições de números inteiros e multiplicações de números inteiros são usadas pelo Algoritmo 1 para multiplicar duas matrizes $n \times n$ com entradas de números inteiros?

Solução: Há n^2 entradas no produto de $\mathbf{A}$ e $\mathbf{B}$. Para encontrar cada entrada, é necessário um total de n multiplicações e $n - 1$ adições. Assim, um total de n^3 multiplicações e $n^2(n - 1)$ adições são usadas. ◀

Surpreendentemente, há algoritmos mais eficientes para multiplicações de matrizes do que o dado no Algoritmo 1. Como mostra o Exemplo 5, para multiplicar duas matrizes $n \times n$ diretamente a partir da definição, são necessárias $O(n^3)$ multiplicações e adições. Usando outros algoritmos, duas matrizes $n \times n$ podem ser multiplicadas usando $O(n^{\sqrt{7}})$ multiplicações e adições. (Mais detalhes sobre tais algoritmos podem ser encontrados em [CoLeRiSt01].)

MULTIPLICAÇÃO ENCADEADA DE MATRIZ Há outro problema importante que envolve a complexidade da multiplicação de matrizes. Como podemos computar a **multiplicação encadeada de matrizes** $\mathbf{A}_1\mathbf{A}_2 \cdots \mathbf{A}_n$ usando o menor número de multiplicações de inteiros, em que $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$ são matrizes $m_1 \times m_2$, $m_2 \times m_3$, $\ldots$, $m_n \times m_{n+1}$, respectivamente, e cada uma tem números inteiros como entradas? (Como a multiplicação de matrizes é associativa, como mostra o Exercício 13 no final desta seção, a ordem das multiplicações usadas não importa.) Antes de estudar este problema, note que as multiplicações $m_1m_2m_3$ de números inteiros são realizadas multiplicando-se uma matriz $m_1 \times m_2$ por uma matriz $m_2 \times m_3$ usando-se o Algoritmo 1 (veja o Exercício 23 no final desta seção). O Exemplo 6 ilustra este problema.

Links

EXEMPLO 6 Em qual ordem as matrizes $\mathbf{A}_1, \mathbf{A}_2$ e $\mathbf{A}_3$ — em que $\mathbf{A}_1$ é 30×20, $\mathbf{A}_2$ é 20×40 e $\mathbf{A}_3$ é 40×10, todas com entradas de inteiros — deveriam ser multiplicadas para usar o menor número de multiplicações de inteiros?

Solução: Há duas maneiras possíveis de computar $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$. Elas são $\mathbf{A}_1(\mathbf{A}_2\mathbf{A}_3)$ e $(\mathbf{A}_1\mathbf{A}_2)\mathbf{A}_3$.

Se $\mathbf{A}_2$ e $\mathbf{A}_3$ forem multiplicadas primeiro, um total de $20 \cdot 40 \cdot 10 = 8000$ multiplicações de inteiros são usadas para obter $\mathbf{A}_2\mathbf{A}_3$, que é uma matriz 20×10. Então, para multiplicar $\mathbf{A}_1$ e $\mathbf{A}_2\mathbf{A}_3$ são necessárias $30 \cdot 20 \cdot 10 = 6000$ multiplicações. Assim, um total de

$$8\,000 + 6\,000 = 14\,000$$

multiplicações é usado. Por outro lado, se $\mathbf{A}_1$ e $\mathbf{A}_2$ forem multiplicadas primeiro, então $30 \cdot 20 \cdot 40 = 24\,000$ multiplicações são usadas para obter a matriz $\mathbf{A}_1\mathbf{A}_2$, que é 30×40. Então, para multiplicar $\mathbf{A}_1\mathbf{A}_2$ e $\mathbf{A}_3$ são necessárias $30 \times 40 \times 10 = 12\,000$ multiplicações. Assim, um total de

$$24\,000 + 12\,000 = 36\,000$$

multiplicações é usado.

Fica claro que o primeiro método é mais eficiente. ◀

Os algoritmos que determinam a maneira mais eficiente para trabalhar com a multiplicação encadeada de matrizes são discutidos em [CoLeRiSt01].

Transposição e Potência de Matrizes

Introduziremos agora uma importante matriz com entradas que sejam zeros e uns.

DEFINIÇÃO 5 A *matriz identidade* de *ordem n* é a matriz $n \times n$ $\mathbf{I}_n = [\delta_{ij}]$, em que $\delta_{ij} = 1$ se $i = j$ e $\delta_{ij} = 0$ se $i \neq j$. Assim,

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Multiplicar uma matriz por uma matriz identidade de tamanho específico, para que exista a multiplicação, não modifica a matriz. Em outras palavras, quando $\mathbf{A}$ é uma matriz $m \times n$, temos

$$\mathbf{A}\mathbf{I}_n = \mathbf{I}_m\mathbf{A} = \mathbf{A}.$$

Potências de matrizes quadradas podem ser definidas. Quando $\mathbf{A}$ é uma matriz $n \times n$, temos

$$\mathbf{A}^0 = \mathbf{I}_n, \qquad \mathbf{A}^r = \underbrace{\mathbf{A}\mathbf{A}\mathbf{A}\cdots\mathbf{A}}_{r \text{ vezes}}.$$

A operação de troca de linhas por colunas de uma matriz quadrada é usada em muitos algoritmos.

DEFINIÇÃO 6 Considere $\mathbf{A} = [a_{ij}]$ como uma matriz $m \times n$. A *transposição* de $\mathbf{A}$, indicada por $\mathbf{A}^t$, é a matriz $n \times m$ obtida pela troca das linhas e colunas de $\mathbf{A}$. Em outras palavras, se $\mathbf{A}^t = [b_{ij}]$, então $b_{ij} = a_{ji}$ para $i = 1, 2, \ldots, n$ e $j = 1, 2, \ldots, m$.

EXEMPLO 7 A transposição da matriz $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ é a matriz $\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$. ◀

As matrizes que permanecem as mesmas quando transpostas normalmente são importantes.

DEFINIÇÃO 7 Uma matriz quadrada $\mathbf{A}$ é chamada de *simétrica* se $\mathbf{A} = \mathbf{A}^t$. Então, $\mathbf{A} = [a_{ij}]$ é simétrica se $a_{ij} = a_{ji}$ para todos os i e j com $1 \leq i \leq n$ e $1 \leq j \leq n$.

Note que uma matriz é simétrica se e somente se for quadrada e simétrica em relação à diagonal principal, que consiste das entradas na *i-ésima* linha e na *i*-ésima coluna para qualquer *i*. Esta simetria é mostrada na Figura 2.

EXEMPLO 8 A matriz $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ é simétrica. ◀

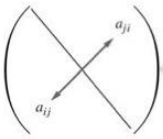


FIGURA 2 Uma Matriz Simétrica.

Matrizes Zero–Um

Uma matriz com entradas que são ou 0 ou 1 é chamada de **matriz zero–um**. Elas são geralmente usadas para representar estruturas discretas, como veremos nos capítulos 8 e 9. Os algoritmos usados nessas estruturas são baseados na aritmética booleana com matrizes zero–um. Essa aritmética é baseada na operações boolenas $\vee$ e $\wedge$, que operam em pares de bits, definidos por

$$b_1 \wedge b_2 = \begin{cases} 1 & \text{se } b_1 = b_2 = 1 \\ 0 & \text{caso contrário,} \end{cases}$$

$$b_1 \vee b_2 = \begin{cases} 1 & \text{se } b_1 = 1 \text{ ou } b_2 = 1 \\ 0 & \text{caso contrário.} \end{cases}$$

DEFINIÇÃO 8 Sejam $\mathbf{A} = [a_{ij}]$ e $\mathbf{B} = [b_{ij}]$ matrizes $m \times n$ zero–um. Então, a *conjunção* de **A** e **B** é a matriz zero–um com (i, j)-ésima entrada $a_{ij} \vee b_{ij}$. A conjunção de **A** e **B** é indicada por $\mathbf{A} \vee \mathbf{B}$. O *encontro* de **A** e **B** é a matriz zero–um com a (i, j)-ésima entrada $a_{ij} \wedge b_{ij}$. O encontro de **A** e **B** é indicado por $\mathbf{A} \wedge \mathbf{B}$.

EXEMPLO 9 Encontre a conjunção e o encontro das matrizes zero–um

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

Solução: Vemos que a conjunção de **A** e **B** é

$$\mathbf{A} \vee \mathbf{B} = \begin{bmatrix} 1 \vee 0 & 0 \vee 1 & 1 \vee 0 \\ 0 \vee 1 & 1 \vee 1 & 0 \vee 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

O encontro de **A** e **B** é

$$\mathbf{A} \wedge \mathbf{B} = \begin{bmatrix} 1 \wedge 0 & 0 \wedge 1 & 1 \wedge 0 \\ 0 \wedge 1 & 1 \wedge 1 & 0 \wedge 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

◀

Definiremos agora o **produto booleano** de duas matrizes.

DEFINIÇÃO 9 Considere $\mathbf{A} = [a_{ij}]$ como uma matriz $m \times k$ zero–um e $\mathbf{B} = [b_{ij}]$ como uma matriz $k \times n$ zero–um. Então, o *produto booleano* de **A** e **B,** indicado por $\mathbf{A} \odot \mathbf{B}$, é a matriz $m \times n$ com a (i, j)-ésima entrada c_{ij}, em que

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \cdots \vee (a_{ik} \wedge b_{kj}).$$

Note que o produto booleano de **A** e **B** é obtido de uma maneira análoga ao produto dessas matrizes, mas com as adições substituídas pela operação $\vee$ e com a multiplicação substituída pela operação $\wedge$. Daremos um exemplo dos produtos booleanos de matrizes.

EXEMPLO 10 Encontre o produto booleano de **A** e **B,** em que

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Solução: O produto booleano $\mathbf{A} \odot \mathbf{B}$ é dado por

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \\ (0 \wedge 1) \vee (1 \wedge 0) & (0 \wedge 1) \vee (1 \wedge 1) & (0 \wedge 0) \vee (1 \wedge 1) \\ (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \end{bmatrix}$$

$$= \begin{bmatrix} 1 \vee 0 & 1 \vee 0 & 0 \vee 0 \\ 0 \vee 0 & 0 \vee 1 & 0 \vee 1 \\ 1 \vee 0 & 1 \vee 0 & 0 \vee 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

◀

O Algoritmo 2 mostra em pseudocódigo a computação do produto booleano de duas matrizes.

ALGORITMO 2 O Produto Booleano.

procedure *produto booleano* (**A**, **B**: matrizes zero–um)
for $i := 1$ **to** m
    for $j := 1$ **to** n
    begin
        $c_{ij} := 0$
        for $q := 1$ **to** k
            $c_{ij} := c_{ij} \vee (a_{iq} \wedge b_{qj})$
    end
{$\mathbf{C} = [c_{ij}]$ é o produto booleano de **A** e **B**}

Podemos também definir as potências booleanas de matrizes quadradas zero–um. Essas potências serão usadas em nossos estudos posteriores sobre temas como caminhos de comunicação em redes de transmissão.

DEFINIÇÃO 10 Considere $\mathbf{A}$ como uma matriz quadrada zero–um e r como um número inteiro positivo. A r-ésima *potência booleana* de $\mathbf{A}$ é o produto booleano de r fatores de $\mathbf{A}$. O r-ésimo produto booleano de $\mathbf{A}$ é indicado por $\mathbf{A}^{[r]}$. Assim,

$$\mathbf{A}^{[r]} = \underbrace{\mathbf{A} \odot \mathbf{A} \odot \mathbf{A} \odot \cdots \odot \mathbf{A}}_{r \text{ vezes}}.$$

(Esta operação é bem definida, pois o produto booleano de matrizes é associativo.) Definimos também $\mathbf{A}^{[0]}$ como sendo $\mathbf{I}_n$.

EXEMPLO 11 Considere $\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$. Encontre $\mathbf{A}^{[n]}$ para todos os números inteiros positivos n.

Solução: Encontramos que

$$\mathbf{A}^{[2]} = \mathbf{A} \odot \mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

Encontramos também que

$$\mathbf{A}^{[3]} = \mathbf{A}^{[2]} \odot \mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \qquad \mathbf{A}^{[4]} = \mathbf{A}^{[3]} \odot \mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Operações adicionais nos mostram que

$$\mathbf{A}^{[5]} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

O leitor pode ver agora que $\mathbf{A}^{[n]} = \mathbf{A}^{[5]}$ para todo número inteiro positivo n com $n \geq 5$. ◀

O número de operações binárias usadas para encontrar o produto booleano de duas matrizes $n \times n$ pode ser facilmente determinado.

EXEMPLO 12 Quantas operações binárias são usadas para encontrar $\mathbf{A} \odot \mathbf{B}$, em que $\mathbf{A}$ e $\mathbf{B}$ são matrizes $n \times n$ zero–um?

Solução: Há n^2 entradas em $\mathbf{A} \odot \mathbf{B}$. Usando o Algoritmo 2, um total de n *OR*s e n *AND*s são usados para encontrar uma entrada de $\mathbf{A} \odot \mathbf{B}$. Assim, $2n$ operações binárias são usadas para encontrar cada entrada. Então, $2n^3$ operações binárias são necessárias para computar $\mathbf{A} \odot \mathbf{B}$ usando o Algoritmo 2. ◀

Exercícios

1. Considere $\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 3 \\ 2 & 0 & 4 & 6 \\ 1 & 1 & 3 & 7 \end{bmatrix}$.

a) Qual o tamanho de $\mathbf{A}$?
b) Qual é a terceira coluna de $\mathbf{A}$?
c) Qual é a segunda linha de $\mathbf{A}$?
d) Qual é o elemento de $\mathbf{A}$ na posição $(3, 2)$?
e) Qual é $\mathbf{A}^t$?

2. Encontre $\mathbf{A} + \mathbf{B}$, em que

a) $\mathbf{A} = \begin{bmatrix} 1 & 0 & 4 \\ -1 & 2 & 2 \\ 0 & -2 & -3 \end{bmatrix}$,

$$\mathbf{B} = \begin{bmatrix} -1 & 3 & 5 \\ 2 & 2 & -3 \\ 2 & -3 & 0 \end{bmatrix}.$$

b) $\mathbf{A} = \begin{bmatrix} -1 & 0 & 5 & 6 \\ -4 & -3 & 5 & -2 \end{bmatrix}$,

$$\mathbf{B} = \begin{bmatrix} -3 & 9 & -3 & 4 \\ 0 & -2 & -1 & 2 \end{bmatrix}.$$

3. Encontre **AB** se

a) $\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 4 \\ 1 & 3 \end{bmatrix}$.

b) $\mathbf{A} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 2 & 3 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 3 & -2 & -1 \\ 1 & 0 & 2 \end{bmatrix}$.

c) $\mathbf{A} = \begin{bmatrix} 4 & -3 \\ 3 & -1 \\ 0 & -2 \\ -1 & 5 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -1 & 3 & 2 & -2 \\ 0 & -1 & 4 & -3 \end{bmatrix}$.

4. Encontre o produto de **AB,** em que

a) $\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & -1 \\ -1 & 1 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 1 & -1 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$.

b) $\mathbf{A} = \begin{bmatrix} 1 & -3 & 0 \\ 1 & 2 & 2 \\ 2 & 1 & -1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 & -1 & 2 & 3 \\ -1 & 0 & 3 & -1 \\ -3 & -2 & 0 & 2 \end{bmatrix}$.

c) $\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 7 & 2 \\ -4 & -3 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 4 & -1 & 2 & 3 & 0 \\ -2 & 0 & 3 & 4 & 1 \end{bmatrix}$.

5. Encontre a matriz **A**, tal que

$$\begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix} \mathbf{A} = \begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix}.$$

[*Dica:* Para encontrar **A** é necessário resolver sistemas de equações lineares.]

6. Encontre a matriz **A**, tal que

$$\begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 1 \\ 4 & 0 & 3 \end{bmatrix} \mathbf{A} = \begin{bmatrix} 7 & 1 & 3 \\ 1 & 0 & 3 \\ -1 & -3 & 7 \end{bmatrix}$$

7. Considere **A** como uma matriz $m \times n$ e **0** como a matriz $m \times n$ que tem todas as entradas iguais a zero. Mostre que $\mathbf{A} = \mathbf{0} + \mathbf{A} = \mathbf{A} + \mathbf{0}$.

8. Mostre que a adição de matrizes é comutativa, ou seja, mostre que se **A** e **B** são matrizes $m \times n$, então $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$.

9. Mostre que a adição de matrizes é associativa, ou seja, mostre que se **A, B** e **C** são matrizes $m \times n$, então $\mathbf{A} + (\mathbf{B} + \mathbf{C}) = (\mathbf{A} + \mathbf{B}) + \mathbf{C}$.

10. Considere **A** como uma matriz 3×4, **B** como uma matriz 4×5 e **C** como uma matriz 4×4. Determine quais dos seguintes produtos são definidos e encontre seus tamanhos.

a) AB **b) BA** **c) AC**
d) CA **e) BC** **f) CB**

11. O que sabemos sobre os tamanhos das matrizes **A** e **B** se ambos os produtos **AB** e **BA** são definidos?

12. Neste exercício, mostraremos que a multiplicação de matrizes é distributiva sobre a adição de matrizes.

a) Suponha que **A** e **B** sejam matrizes $m \times k$ e que **C** seja uma matriz $k \times n$. Mostre que $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$.

b) Suponha que **C** seja uma matriz $m \times k$ e que **A** e **B** sejam matrizes $k \times n$. Mostre que $\mathbf{C}(\mathbf{A} + \mathbf{B}) = \mathbf{CA} + \mathbf{CB}$.

13. Neste exercício, mostraremos que a multiplicação de matrizes é associativa. Suponha que **A** seja uma matriz $m \times p$, **B** seja uma matriz $p \times k$ e **C** seja uma matriz $k \times n$. Mostre que $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$.

14. A matriz $n \times n$ $\mathbf{A} = [a_{ij}]$ é chamada de **matriz diagonal** se $a_{ij} = 0$ quando $i \neq j$. Mostre que o produto de duas matrizes diagonais $n \times n$ é novamente uma matriz diagonal. Dê uma regra simples para determinar este produto.

15. Considere

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Encontre uma fórmula para $\mathbf{A}^n$, sempre que n for um número inteiro positivo.

16. Mostre que $(\mathbf{A}^t)^t = \mathbf{A}$.

17. Considere **A** e **B** como duas matrizes $n \times n$. Mostre que

a) $(\mathbf{A} + \mathbf{B})^t = \mathbf{A}^t + \mathbf{B}^t$.

b) $(\mathbf{AB})^t = \mathbf{B}^t\mathbf{A}^t$.

Se **A** e **B** forem matrizes $n \times n$ com $\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n$, então **B** é chamada de **inversa** de **A** (esta terminologia é apropriada porque tal matriz **B** é única) e **A** é dita **inversível.** A notação $\mathbf{B} = \mathbf{A}^{-1}$ indica que **B** é a matriz inversa de **A.**

18. Mostre que

$$\begin{bmatrix} 2 & 3 & -1 \\ 1 & 2 & 1 \\ -1 & -1 & 3 \end{bmatrix}$$

é a inversa de

$$\begin{bmatrix} 7 & -8 & 5 \\ -4 & 5 & -3 \\ 1 & -1 & 1 \end{bmatrix}.$$

19. Considere **A** como a matriz 2×2

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Mostre que se $ad - bc \neq 0$, então

$$\mathbf{A}^{-1} = \begin{bmatrix} \dfrac{d}{ad-bc} & \dfrac{-b}{ad-bc} \\ \dfrac{-c}{ad-bc} & \dfrac{a}{ad-bc} \end{bmatrix}.$$

20. Considere

$$\mathbf{A} = \begin{bmatrix} -1 & 2 \\ 1 & 3 \end{bmatrix}.$$

a) Encontre $\mathbf{A}^{-1}$. [*Dica:* Use o Exercício 19.]
b) Encontre $\mathbf{A}^3$.
c) Encontre $(\mathbf{A}^{-1})^3$.
d) Use suas respostas dos itens (b) e (c) para mostrar que $(\mathbf{A}^{-1})^3$ é a matriz inversa de $\mathbf{A}^3$.

21. Seja **A** uma matriz inversível. Mostre que $(\mathbf{A}^n)^{-1} = (\mathbf{A}^{-1})^n$ sempre que n for um número inteiro positivo.

22. Considere **A** como uma matriz. Mostre que a matriz $\mathbf{AA}^t$ é simétrica. [*Dica:* Mostre que esta matriz é igual à sua transposta com a ajuda do Exercício 17b.]

23. Mostre que o algoritmo convencional usa $m_1m_2m_3$ multiplicações para computar o produto da matriz **A** $m_1 \times m_2$ e da matriz **B** $m_2 \times m_3$.

24. Qual é a maneira mais eficiente de multiplicar as matrizes $\mathbf{A}_1$, $\mathbf{A}_2$ e $\mathbf{A}_3$ com os tamanhos abaixo
a) 20×50, 50×10, 10×40?
b) 10×5, 5×50, 50×1?

25. Qual é a maneira mais eficiente de multiplicar as matrizes $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ e $\mathbf{A}_4$ se as dimensões dessas matrizes são 10×2, 2×5, 5×20 e 20×3, respectivamente?

26. **a)** Mostre que o sistema de equações lineares simultâneas

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

nas variáveis $x_1, x_2, \ldots, x_n$ pode ser expresso como $\mathbf{AX} = \mathbf{B}$, em que $\mathbf{A} = [a_{ij}]$, $\mathbf{X}$ é uma matriz $n \times 1$ com x_i a entrada na i-ésima linha, e $\mathbf{B}$ é uma matriz $n \times 1$ com b_i a entrada na sua i-ésima linha.

b) Mostre que se a matriz $\mathbf{A} = [a_{ij}]$ é inversível (como definido no preâmbulo do Exercício 18), então a solução do sistema no item (a) pode ser encontrada usando-se a equação $\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$.

27. Use os exercícios 18 e 26 para resolver o sistema

$$\begin{aligned} 7x_1 - 8x_2 + 5x_3 &= 5 \\ -4x_1 + 5x_2 - 3x_3 &= -3 \\ x_1 - x_2 + x_3 &= 0 \end{aligned}$$

28. Considere

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{B} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Encontre
a) $\mathbf{A} \vee \mathbf{B}$. **b)** $\mathbf{A} \wedge \mathbf{B}$. **c)** $\mathbf{A} \odot \mathbf{B}$.

29. Considere

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{B} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

Encontre
a) $\mathbf{A} \vee \mathbf{B}$. **b)** $\mathbf{A} \wedge \mathbf{B}$. **c)** $\mathbf{A} \odot \mathbf{B}$.

30. Encontre o produto booleano de **A** e **B**, em que

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

31. Considere

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Encontre
a $\mathbf{A}^{[2]}$. **b)** $\mathbf{A}^{[3]}$.
c) $\mathbf{A} \vee \mathbf{A}^{[2]} \vee \mathbf{A}^{[3]}$.

32. Considere **A** como uma matriz zero–um. Mostre que
a) $\mathbf{A} \vee \mathbf{A} = \mathbf{A}$.
b) $\mathbf{A} \wedge \mathbf{A} = \mathbf{A}$.

33. Neste exercício mostraremos que as operações de encontro e de conjunção são comutativas. Considere **A** e **B** como matrizes $m \times n$ zero–um. Mostre que
a) $\mathbf{A} \vee \mathbf{B} = \mathbf{B} \vee \mathbf{A}$.
b) $\mathbf{B} \wedge \mathbf{A} = \mathbf{A} \wedge \mathbf{B}$.

34. Neste exercício mostraremos que as operações de encontro e de conjunção são associativas. Considere **A**, **B** e **C** como matrizes $m \times n$ zero–um. Mostre que
a) $(\mathbf{A} \vee \mathbf{B}) \vee \mathbf{C} = \mathbf{A} \vee (\mathbf{B} \vee \mathbf{C})$.
b) $(\mathbf{A} \wedge \mathbf{B}) \wedge \mathbf{C} = \mathbf{A} \wedge (\mathbf{B} \wedge \mathbf{C})$.

35. Vamos estabelecer as propriedades distributivas das operações de encontro sobre as de conjunção neste exercício. Considere **A**, **B** e **C** como matrizes $m \times n$ zero–um. Mostre que
a) $\mathbf{A} \vee (\mathbf{B} \wedge \mathbf{C}) = (\mathbf{A} \vee \mathbf{B}) \wedge (\mathbf{A} \vee \mathbf{C})$.
b) $\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C}) = (\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{C})$.

36. Considere **A** como uma matriz $n \times n$ zero–um e **I** como a matriz identidade $n \times n$. Mostre que $\mathbf{A} \odot \mathbf{I} = \mathbf{I} \odot \mathbf{A} = \mathbf{A}$.

37. Neste exercício mostraremos que o produto booleano de matrizes zero–um é associativo. Suponha que **A** seja uma matriz $m \times p$ zero–um, **B** seja uma matriz $p \times k$ zero–um e **C** seja uma matriz $k \times n$ zero–um. Mostre que $\mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}) = (\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C}$.

Termos-chave e Resultados

TERMOS

algoritmo: um conjunto finito de instruções precisas para realizar uma computação ou resolver um problema

algoritmo de busca: o problema de localizar um elemento em uma lista

algoritmo de busca linear: um procedimento para localizar em uma lista elemento por elemento

algoritmo de busca binária: um procedimento para localizar um elemento em uma lista ordenada por meio de substituições sucessivas na lista, que é dividida pela metade

ordenação: a reordenação de elementos em uma lista em ordem não decrescente

algoritmo "voraz": um algoritmo que faz a melhor escolha em cada etapa

$f(x)$ é $O(g(x))$: o fato de que $|f(x)| \leq C|g(x)|$ para todo $x > k$ para algumas constantes C e k

parâmetros da relação de $f(x)$ é $O(g(x))$: um par C e k, tal que $|f(x)| \leq C|g(x)|$ sempre que $x > k$

$f(x)$ é $\Omega(g(x))$: o fato de que $|f(x)| \geq C|g(x)|$ para todo $x > k$ para algumas constantes positivas C e k

$f(x)$ é $\Theta(g(x))$: o fato de que $f(x)$ é $O(g(x))$ e $f(x)$ é $\Omega(g(x))$

complexidade temporal: a quantidade de tempo necessária para um algoritmo resolver um problema

complexidade espacial: a quantidade de espaço de armazenagem necessária para um algoritmo resolver um problema

complexidade temporal de pior caso: a maior quantidade de tempo necessária para um algoritmo resolver um problema de determinado tamanho

complexidade temporal de caso médio: a média de tempo necessário para um algoritmo resolver um problema de um dado tamanho

$a \mid b$ (a divide b): há um número inteiro c tal que $b = ac$

número primo: um número inteiro positivo maior que 1 com exatamente dois divisores inteiros positivos

número composto: um número inteiro positivo maior que 1 que não é primo

primo de Mersenne: um número primo na forma $2^p - 1$, em que p é primo

mdc(a, b) (máximo divisor comum de a e b): o maior número inteiro que divide a e b

primos entre si (ou relativamente primos): números inteiros a e b, tal que $\text{mdc}(a, b) = 1$

par de inteiros primos entre si: um conjunto de números inteiros com a propriedade de que cada par desses inteiros é relativamente primo ou primo entre si

mmc(a, b) (mínimo múltiplo comum de a e b): o menor número inteiro positivo que é divisível por a e b

a mod b: o resto quando o número inteiro a é dividido pelo inteiro positivo b

$a \equiv b \pmod{m}$ (a é congruente a b módulo m): $a - b$ é divisível por m

codificação: o processo de tornar uma mensagem secreta

decodificação: o processo de retornar a mensagem secreta à sua forma original

$n = (a_k a_{k-1} \ldots a_1 a_0)_b$: a representação de n na base b

representação binária: a representação na base 2 de um número inteiro

representação hexadecimal: a representação na base 16 de um número inteiro

representação octal: a representação na base 8 de um número inteiro

combinação linear de a e b com coeficientes inteiros: um número na forma $sa + tb$, em que s e t são números inteiros

inverso de a módulo m: um número inteiro $\bar{a}$ tal que $\bar{a}a \equiv 1 \pmod{m}$

congruência linear: uma congruência na forma $ax \equiv b \pmod{m}$, em que x é uma variável

pseudoprimo na base 2: um número inteiro composto n tal que $2^{n-1} \equiv 1 \pmod{n}$

pseudoprimo na base b: um número inteiro composto n tal que $b^{n-1} \equiv 1 \pmod{n}$

número de Carmichael: um número inteiro composto n tal que n é um pseudoprimo de base b para todos os números inteiros positivos b com $\text{mdc}(b, n) = 1$

codificação de chave privada: codificação em que ambas as chaves de codificação e decodificação devem ser mantidas em segredo

codificação de chave pública: codificação em que as chaves de codificação são de acesso público, mas as de decodificação são mantidas em segredo

matriz: uma disposição retangular de números

adição de matrizes: veja a página 247

multiplicação de matrizes: veja a página 248

I_n (matriz identidade de ordem n): a matriz $n \times n$ que tem entradas iguais a 1 em sua diagonal e 0 nas demais posições

$\mathbf{A}^t$ (transposta de A): a matriz obtida a partir de $\mathbf{A}$ pela troca das linhas por colunas

matriz simétrica: uma matriz é simétrica se ela for igual à sua transposta

matriz zero–um: uma matriz com cada entrada igual a 0 ou a 1

$\mathbf{A} \vee \mathbf{B}$ (a conjunção de A e B): veja a página 252

$\mathbf{A} \wedge \mathbf{B}$ (o encontro de A e B): veja a página 252

$\mathbf{A} \odot \mathbf{B}$ (o produto booleano de A e B): veja a página 253

RESULTADOS

algoritmos de busca linear e binário: (dados na Seção 3.1)

bubble sort: uma ordenação que usa passos em que itens sucessivos são trocados se estiverem fora de ordem

insertion sort: uma ordenação em que na j-ésima etapa é inserido o j-ésimo elemento na posição correta na lista dos primeiros $j - 1$ elementos ordenados

A busca linear tem complexidade $O(n)$.

A busca binária tem complexidade $O(\log n)$.

O *bubble sort* e o *insertion sort* têm complexidade $O(n^2)$.

$\log n!$ é $O(n \log n)$.

Se $f_1(x)$ é $O(g_1(x))$ e $f_2(x)$ é $O(g_2(x))$, então $(f_1 + f_2)(x)$ é $O(\max(g_1(x), g_2(x)))$ e $(f_1 f_2)(x)$ é $O(g_1(x)\, g_2(x))$.

Se $a_0, a_1, \ldots, a_n$ são números reais com $a_n \neq 0$, então $a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ é $O(x^n)$ e $\Theta(x^n)$.

Teorema Fundamental da Aritmética: Todo número inteiro positivo pode ser escrito unicamente como o produto de números primos, em que os fatores dos primos são escritos em ordem crescente de tamanho.

algoritmo de divisão: Considere a e d como números inteiros com d positivo. Então, há números inteiros únicos q e r com $0 \leq r < d$, tal que $a = dq + r$.

Se a e b forem números inteiros positivos, então $ab = \text{mdc}(a, b) \cdot \text{mmc}(a, b)$.

algoritmo de Euclides: para encontrar o máximo divisor comum (veja o Algoritmo 6 na Seção 3.6)

Considere b como um número inteiro positivo maior que 1. Então, se n é um número inteiro positivo, n pode ser expresso unicamente na forma $n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$.

O algoritmo para encontrar a expansão de base b de um número inteiro (veja o Algoritmo 1 na Seção 3.6)

Os algoritmos convencionais para adição e multiplicação de números inteiros (dados na Seção 3.6)

O algoritmo de potenciação modular (veja o Algoritmo 5 na Seção 3.6)

O máximo divisor comum de dois números inteiros pode ser expresso como uma combinação linear com coeficientes inteiros desses números.

Se m for um número inteiro positivo e $\text{mdc}(a, m) = 1$, então a tem um único inverso módulo m.

Teorema Chinês do Resto: Um sistema de congruências lineares módulo pares de primos inteiros relativos tem uma única solução módulo produto destes módulos.

Pequeno Teorema de Fermat: Se p é número primo e $p \nmid a$, então $a^{p-1} \equiv 1 \pmod{p}$.

Questões de Revisão

1. a) Defina o termo *algoritmo*.
b) Quais são as diferentes maneiras de descrever algoritmos?
c) Qual é a diferença entre um algoritmo que resolve um problema e um programa de computação que resolve esse mesmo problema?

2. a) Descreva, usando a língua portuguesa, um algoritmo que encontre o maior número inteiro em uma lista de n inteiros.
b) Expresse esse algoritmo em pseudocódigo.
c) Quantas comparações o algoritmo usa?

3. a) Defina o fato de que $f(n)$ é $O(g(n))$, em que $f(n)$ e $g(n)$ são funções do conjunto dos números inteiros positivos para o conjunto dos números reais.
b) Use a definição do fato de que $f(n)$ é $O(g(n))$ diretamente para demonstrar ou negar que $n^2 + 18n + 107$ é $O(n^3)$.
c) Use a definição do fato de que $f(n)$ é $O(g(n))$ diretamente para demonstrar ou negar que n^3 é $O(n^2 + 18n + 107)$.

4. a) Como você pode produzir uma estimativa O grande para uma função que é a soma de termos diferentes, em que cada termo é o produto de várias funções?
b) Dê uma estimativa O grande para a função $f(n) = (n! + 1)(2^n + 1) + (n^{n-2} + 8n^{n-3})(n^3 + 2^n)$. Para a função g em sua estimativa $f(x)$ é $O(g(x))$, use uma função simples de menor ordem possível.

5. a) Defina o que significa a complexidade temporal de pior caso, a complexidade temporal de caso médio e a complexidade temporal de melhor caso (em termos de comparações) para um algoritmo que encontra o menor número inteiro em uma lista de n inteiros.
b) Quais são as complexidades temporais de pior caso, de caso médio e de melhor caso, em termos de comparações, do algoritmo que encontra o menor número inteiro em uma lista de n inteiros comparando cada um deles com o menor número inteiro encontrado até aqui?

6. a) Descreva o algoritmo de busca linear e o de busca binária para encontrar um número inteiro em uma lista de inteiros em ordem crescente.
b) Compare as complexidades temporais de pior caso desses dois algoritmos.
c) Um desses algoritmos é sempre mais rápido que o outro (medindo em termos de comparações)?

7. a) Descreva o algoritmo bubble sort.
b) Use o algoritmo bubble sort para ordenar a lista 5, 2, 4, 1, 3.
c) Dê uma estimativa O grande para o número de comparações usadas pelo bubble sort.

8. a) Descreva o algoritmo insertion sort.
b) Use o algoritmo insertion sort para ordenar a lista 2, 5, 1, 4, 3.
c) Dê uma estimativa O grande para o número de comparações usadas pelo insertion sort.

9. a) Explique o conceito de algoritmo voraz.
b) Forneça um exemplo de um algoritmo voraz que produza uma solução ótima e explique por que ele faz isso.
c) Forneça um exemplo de algoritmo voraz que nem sempre produz uma solução ótima e explique por que isso ocorre.

10. Enuncie o Teorema Fundamental da Aritmética.

11. a) Descreva um procedimento para encontrar a fatoração em números primos de um número inteiro.
b) Use esse procedimento para encontrar a fatoração em números primos de 80 707.

12. a) Defina o máximo divisor comum de dois números inteiros.
b) Descreva, pelo menos, três maneiras diferentes de encontrar o máximo divisor comum de dois números inteiros. Quando cada um dos métodos funciona melhor?
c) Encontre o máximo divisor comum de 1 234 567 e 7 654 321.
d) Encontre o máximo divisor comum de $2^3 3^5 5^7 7^9 11$ e $2^9 3^7 5^5 7^3 13$.

13. a) Defina o que significa a e b serem congruentes módulo 7.
b) Quais pares de números inteiros -11, -8, -7, -1, 0, 3 e 17 são congruentes módulo 7?

c) Mostre que se a e b são congruentes módulo 7, então $10a + 13$ e $-4b + 20$ são também congruentes módulo 7.

14. Descreva um procedimento para converter expansões decimais (base 10) de números inteiros em expansões hexadecimais.

15. a) Como você pode encontrar uma combinação linear (com coeficientes inteiros) de dois números inteiros que seja igual ao seu máximo divisor comum?

b) Expresse mdc(84, 119) como uma combinação linear de 84 e 119.

16. a) O que significa para $\overline{a}$ ser o inverso de a módulo m?

b) Como você pode encontrar um inverso de a módulo m quando m é um número inteiro positivo e $\text{mdc}(a, m) = 1$?

c) Encontre um inverso de 7 módulo 19.

17. a) Como um inverso de a módulo m pode ser usado para resolver a congruência $ax \equiv b \pmod{m}$ quando mdc $(a, m) = 1$?

b) Resolva a congruência linear $7x \equiv 13 \pmod{19}$.

18. a) Enuncie o Teorema Chinês do Resto.

b) Encontre as soluções para o sistema $x \equiv 1 \pmod 4$, $x \equiv 2 \pmod 5$ e $x \equiv 3 \pmod 7$.

19. Suponha que $2^{n-1} \equiv 1 \pmod{n}$. n é necessariamente primo?

20. a) Qual é a diferença entre um criptossistema de chave pública e um de chave privada?

b) Explique por que um sistema de chave privada utiliza códigos de substituição.

c) Explique por que o sistema RSA é de chave pública.

21. Defina o produto de duas matrizes **A** e **B**. Quando seu produto é definido?

22. a) Quantas maneiras diferentes existem para avaliar o produto $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4$ pela multiplicação sucessiva de pares de matrizes, quando esse produto é definido?

b) Suponha que $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ e $\mathbf{A}_4$ sejam matrizes 10×20, 20×5, 5×10 e 10×5, respectivamente. Como $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4$ deveria ser computada usando-se o menor número de multiplicações com as entradas?

Exercícios Complementares

1. a) Descreva um algoritmo para localizar a última ocorrência de maior número em uma lista de números inteiros.

b) Faça a estimativa do número de comparações usadas.

2. a) Descreva um algoritmo para encontrar o primeiro e o segundo maiores elementos em uma lista de números inteiros.

b) Faça a estimativa do número de comparações usadas.

3. a) Dê um algoritmo para determinar se uma cadeia de bits contém um par de zeros consecutivos.

b) Quantas comparações o algoritmo usa?

4. a) Suponha que uma lista tenha números inteiros que estão em ordem do maior para o menor e um número inteiro possa aparecer repetidamente nesta lista. Construa um algoritmo que localize todas as ocorrências de um número inteiro x nesta lista.

b) Faça a estimativa do número de comparações usadas.

5. a) Adapte o Algoritmo 1 da Seção 3.1 para encontrar o máximo e o mínimo em uma seqüência de n elementos empregando um máximo temporário e um mínimo temporário que sejam atualizados à medida que cada elemento sucessivo for examinado.

b) Descreva o algoritmo do item (a) em pseudocódigo.

c) Quantas comparações de elementos na seqüência são realizadas por esse algoritmo? (Não conte as comparações usadas para determinar se o final da seqüência foi alcançado.)

6. a) Descreva em detalhes os passos de um algoritmo que encontra o máximo e o mínimo em uma seqüência de n elementos examinando pares de elementos sucessivos, mantendo um máximo temporário e um mínimo temporário. Se n for ímpar, o máximo e o mínimo temporários deverão ser inicialmente iguais ao primeiro termo e, se n for par, o máximo e o mínimo temporários deverão ser encontrados comparando-se os dois elementos iniciais. O máximo e o mínimo temporários deverão ser atualizados comparando-os com o máximo e o mínimo dos pares de elementos a ser examinados.

b) Expresse o algoritmo descrito no item (a) em pseudocódigo.

c) Quantas comparações de elementos da seqüência são realizadas por esse algoritmo? (Não conte as comparações usadas para determinar se o final da seqüência foi alcançado.) Como comparar isso com o número de comparações usadas pelo algoritmo no Exercício 5?

***7.** Mostre que a complexidade de pior caso em termos de comparações de um algoritmo que encontra o máximo e o mínimo de n elementos é, pelo menos, $\lceil 3n/2 \rceil - 2$.

8. Construa um algoritmo eficiente para encontrar o segundo maior elemento em uma seqüência de n elementos e determine a complexidade de pior caso do seu algoritmo.

O **shaker sort** (ou **bubble sort bidirecional**) compara sucessivamente pares adjacentes de elementos, mudando-os se eles estiverem fora de ordem e passando pela lista alternadamente desde o começo até o fim, e então do fim até o começo, até que nenhuma mudança mais seja necessária.

9. Mostre os passos usados pelo shaker sort para ordenar a lista 3, 5, 1, 4, 6, 2.

10. Expresse o shaker sort em pseudocódigo.

11. Mostre que o shaker sort tem complexidade $O(n^2)$ medida em termos do número de comparações que ele usa.

12. Explique por que o shaker sort é eficiente para ordenar listas que já estão próximas de sua ordem correta.

13. Mostre que $(n \log n + n^2)^3$ é $O(n^6)$.

14. Mostre que $8x^3 + 12x + 100 \log x$ é $O(x^3)$.

15. Dê uma estimativa O grande para $(x^2 + x(\log x)^3) \cdot (2^x + x^3)$.

16. Encontre a estimativa O grande para $\sum_{j=1}^{n} j(j+1)$.

***17.** Mostre que $n!$ não é $O(2^n)$.

***18.** Mostre que n^n não é $O(n!)$.

19. Encontre quatro números congruentes a 5 módulo 17.

20. Mostre que se a e d forem números inteiros positivos, então existem inteiros q e r, tal que $a = dq + r$, em que $-d/2 < r \leq d/2$.

***21.** Mostre que se $ac \equiv bc \pmod{m}$, então $a \equiv b \pmod{m/d}$, em que $d = \text{mdc}(m, c)$.

***22.** Quantos zeros estão presentes no final da expansão binária de $100_{10}!$?

23. Use o algoritmo de Euclides para encontrar o máximo divisor comum de 10 223 e 33 341.

24. Quantos divisões são necessárias para encontrar mdc(144, 233) usando-se o algoritmo de Euclides?

25. Encontre mdc$(2n + 1, 3n + 2)$, em que n é um número inteiro positivo. [*Dica:* Use o algoritmo de Euclides.]

26. **a)** Mostre que se a e b forem números inteiros positivos com $a \geq b$, então mdc$(a, b) = a$ se $a = b$, mdc$(a, b) = 2$ mdc$(a/2, b/2)$ se a e b forem pares, mdc$(a, b) =$ mdc$(a/2, b)$ se a for par e b for ímpar, e mdc$(a, b) =$ mdc$(a - b, b)$ se a e b forem ímpares.

b) Explique como usar o item (a) para construir um algoritmo que possa computar o máximo divisor comum de dois números inteiros positivos que use apenas comparações, subtrações e substituições de expansões binárias, sem usar qualquer divisão.

c) Encontre mdc(1202, 4848) usando esse algoritmo.

27. Mostre que um número inteiro é divisível por 9 se e somente se a soma de seus dígitos decimais for divisível por 9.

28. Demonstre que se n for um número inteiro positivo, tal que a soma dos divisores de n é $n + 1$, então n é um número primo.

29. Adapte a demonstração de que há um número infinito de primos (Teorema 3 da Seção 3.5) para mostrar que há um número infinito de primos na forma $6k + 5$, em que k é um número inteiro positivo.

30. **a)** Explique por que n **div** 7 é igual ao número de semanas em n dias.

b) Explique por que n **div** 24 é igual ao número de dias em n horas.

Um conjunto de números inteiros é chamado de **primos relativos mutuamente** se o máximo divisor comum desses inteiros for 1.

31. Determine se estes conjuntos de números inteiros são primos relativos mutuamente.

a) 8, 10, 12 **b)** 12, 15, 25

c) 15, 21, 28 **d)** 21, 24, 28, 32

32. Encontre um conjunto de quatro primos relativos mutuamente, tal que não sejam primos entre si.

33. **a)** Suponha que certas mensagens sejam codificadas usando-se a função $f(p) = (ap + b)$ **mod** 26, tal que mdc$(a, 26) = 1$. Determine uma função que pode ser usada para decodificar as mensagens.

b) A versão codificada de uma mensagem é LJMKG MGMXF QEXMW. Supondo que ela foi codificada usando-se a função $f(p) = (7p + 10)$ **mod** 26, qual era a mensagem original?

34. Mostre que o sistema de congruências $x \equiv 2 \pmod 6$ e $x \equiv 3 \pmod 9$ não tem soluções.

35. Encontre todas as soluções do sistema de congruências $x \equiv 4 \pmod 6$ e $x \equiv 13 \pmod{15}$.

***36.** **a)** Mostre que o sistema de congruências $x \equiv a_1 \pmod{m_1}$ e $x \equiv a_2 \pmod{m_2}$ tem uma solução se e somente se $\text{MDC}(m_1, m_2) \mid a_1 - a_2$.

b) Mostre que a solução do item (a) é único módulo mmc(m_1, m_2).

***37.** Demonstre que se $f(x)$ for um polinômio não constante com coeficientes inteiros, então há um número inteiro y tal que $f(y)$ seja um número composto. [*Dica:* Suponha que $f(x_0) = p$ seja um número primo. Mostre que p divide $f(x_0 + kp)$ para todos os números inteiros k. Obtenha uma contradição do fato de que um polinômio de grau n, em que $n > 1$, tenha cada valor no máximo n vezes.]

***38.** Mostre que se a e b forem números irracionais positivos, tal que $1/a + 1/b = 1$, então todo número inteiro positivo pode ser unicamente expresso por $\lfloor ka \rfloor$ ou $\lfloor kb \rfloor$ para qualquer número inteiro positivo k.

39. Mostre que a conjectura de Goldbach, que afirma que todo número inteiro par maior que 2 é a soma de dois primos, é equivalente à afirmação de que todo número inteiro maior que 5 é a soma de três primos.

40. Demonstre que não existem soluções em números inteiros x e y para a equação $x^2 - 5y^2 = 2$. [*Dica:* Considere esta equação módulo 5.]

41. Encontre $\mathbf{A}^n$ se $\mathbf{A}$ for

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

42. Mostre que se $\mathbf{A} = c\mathbf{I}$, em que c é um número real e $\mathbf{I}$ é a matriz identidade $n \times n$, então $\mathbf{AB} = \mathbf{BA}$ sempre que $\mathbf{B}$ for uma matriz $n \times n$.

43. Mostre que se $\mathbf{A}$ for uma matriz 2×2 , tal que $\mathbf{AB} = \mathbf{BA}$ para toda matriz $\mathbf{B}$ 2×2, então $\mathbf{A} = c\mathbf{I}$, em que c é um número real e $\mathbf{I}$ é a matriz identidade 2×2.

Uma matriz $n \times n$ é chamada de **triangular** se $a_{ij} = 0$ sempre que $i > j$.

44. A partir da definição de produto de matrizes, construa um algoritmo para computar o produto de duas matrizes triangulares que ignore aqueles produtos na computação que são automaticamente iguais a zero.

45. Dê uma descrição em pseudocódigo do algoritmo do Exercício 44 para multiplicar duas matrizes triangulares.

46. Quantas multiplicações de entradas são usadas pelo algoritmo encontrado no Exercício 44 para multiplicar duas matrizes triangulares $n \times n$?

47. Mostre que se $\mathbf{A}$ e $\mathbf{B}$ forem matrizes inversíveis e $\mathbf{AB}$ existir, então $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.

48. Qual a melhor ordem de fazer o produto $\mathbf{ABCD}$ se $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ e $\mathbf{D}$ forem matrizes com dimensões 30×10, 10×40, 40×50 e 50×30, respectivamente? Suponha que o nú-

mero de multiplicações de entradas usadas para multiplicar uma matriz $p \times q$ e uma matriz $q \times r$ seja pqr.

49. Considere **A** como uma matriz $n \times n$ e **0** como a matriz $n \times n$ com todas as entradas iguais a zero. Mostre que as seguintes operações são verdadeiras.
a) $\mathbf{A} \odot \mathbf{0} = \mathbf{0} \odot \mathbf{A} = \mathbf{0}$ **b)** $\mathbf{A} \vee \mathbf{0} = \mathbf{0} \vee \mathbf{A} = \mathbf{A}$
c) $\mathbf{A} \wedge \mathbf{0} = \mathbf{0} \wedge \mathbf{A} = \mathbf{0}$

50. Mostre que se as denominações de moedas são $c^0, c^1, \ldots, c^k$, em que k é um número inteiro positivo e c é um número inteiro positivo, $c > 1$, o algoritmo voraz sempre produz trocos usando o menor número de moedas possível.

51. Construa um algoritmo para encontrar um número entre 1 e $2^n - 1$ procurando sucessivamente por cada bit em sua expansão binária.

52. Determine a complexidade, em termos do número de tentativas, necessária para determinar um número entre 1 e $2^n - 1$ procurando sucessivamente por todos os bits em sua expansão binária.

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dada uma lista de n números inteiros, encontre o maior inteiro na lista.

2. Dada uma lista de n números inteiros, encontre a primeira e a última ocorrência do maior inteiro na lista.

3. Dada uma lista de n números inteiros distintos, determine a posição de um inteiro na lista usando uma busca linear.

4. Dada uma lista ordenada de n números inteiros distintos, determine a posição de um inteiro na lista usando uma busca binária.

5. Dada uma lista de n números inteiros, ordene-os usando o bubble sort.

6. Dada uma lista de n números inteiros, ordene-os usando o insertion sort.

7. Dado um número inteiro n, use o algoritmo voraz para encontrar o troco para n centavos usando moedas de 25, 20, 5 e 1 centavo.

8. Dada uma lista ordenada de n números inteiros e um inteiro x, encontre o número de comparações usadas para determinar a posição de um inteiro na lista usando uma busca linear e usando uma busca binária.

9. Dada uma lista de números inteiros, determine o número de comparações usadas pelo bubble sort e pelo insertion sort para ordenar essa lista.

10. Dado um número inteiro positivo, determine se ele é primo.

11. Dada uma mensagem, codifique-a usando o código de César, e dada uma mensagem codificada pelo código de César, decodifique-a.

12. Dados dois números inteiros positivos, encontre seu máximo divisor comum usando o algoritmo de Euclides.

13. Dados dois números inteiros positivos, encontre seu mínimo múltiplo comum.

***14.** Dado um número inteiro positivo, encontre a fatoração em números primos deste inteiro.

15. Dado um número inteiro positivo e um inteiro positivo b maior que 1, encontre a expansão na base b deste inteiro.

16. Dados os números inteiros positivos a, b e $m > 1$, encontre a^b **mod** m.

17. Dado um número inteiro positivo, encontre a expansão de Cantor desse inteiro (veja o preâmbulo do Exercício 46 da Seção 3.6).

18. Dado um número inteiro positivo n, um módulo m, multiplicador a, um acréscimo c e uma origem x_0, com $0 \leq a < m$, $0 \leq c < m$ e $0 \leq x_0 < m$, construa a seqüência de n números pseudo-aleatórios usando o gerador de congruência linear $x_{n+1} = (ax_n + c)$ **mod** m.

19. Dados os números inteiros positivos a e b, encontre os inteiros s e t, tal que $sa + tb = \text{mdc}(a, b)$.

20. Dadas as n congruências lineares com módulos de pares de primos entre si, encontre a solução simultânea dessas congruências módulo produto desses módulos.

21. Dada uma matriz **A** $m \times k$ e uma matriz **B** $k \times n$, encontre **AB**.

22. Dada uma matriz quadrada **A** e um número inteiro positivo n, encontre $\mathbf{A}^n$.

23. Dada uma matriz quadrada, determine se ela é simétrica.

24. Dada uma matriz **A** $n_1 \times n_2$, uma matriz **B** $n_2 \times n_3$, uma matriz **C** $n_3 \times n_4$ e uma matriz **D** $n_4 \times n_5$, todas com elementos inteiros, determine a ordem mais eficiente para multiplicar essas matrizes (em termos de números de multiplicações e adições de inteiros).

25. Dadas duas matrizes booleanas $m \times n$, descubra seu encontro e sua conjunção.

26. Dada uma matriz booleana **A** $m \times k$ e uma matriz booleana **B** $k \times n$, encontre o produto booleano de **A** e **B**.

27. Dada uma matriz booleana quadrada **A** e um número inteiro positivo n, encontre $\mathbf{A}^{[n]}$.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Sabemos que n^b é $O(d^n)$ quando b e d são números inteiros positivos com $d \geq 2$. Dê os valores das constantes C e k, tal que $n^b \leq Cd^n$ sempre que $x > k$ para cada um desses conjuntos de valores: $b = 10$, $d = 2$; $b = 20$, $d = 3$; $b = 1000$, $d = 7$.

2. Compute o troco para diferentes valores de n, com moedas de diferentes denominações, usando o algoritmo voraz e determine se o menor número de moedas foi usado. Você pode encontrar as condições para que o algoritmo voraz garanta o uso do menor número de moedas possível?
3. Usando um gerador de ordenação aleatória dos números inteiros 1, 2, . . . , n, encontre o número de comparações usadas pelo bubble sort, insertion sort, insertion sort binário e selection sort para ordenar esses inteiros.
4. Determine se $2^p - 1$ é primo para cada um dos primos não excedentes a 100.
5. Teste uma quantia de números de Mersenne grandes $2^p - 1$ para determinar se eles são primos. (Você pode usar o software do projeto GIMPS.)
6. Procure por polinômios com uma variável cujos valores de execução longa de números inteiros consecutivos sejam todos primos.
7. Encontre quantos primos você puder na forma $n^2 + 1$, em que n é um número inteiro positivo. Não sabemos se existe um número infinito de tais primos.
8. Encontre 10 números primos diferentes, cada um com 100 dígitos.
9. Quantos números primos existem menores que 1 000 000, menores que 10 000 000 e menores que 100 000 000? Você pode propor uma estimativa para o número de primos menores que x, em que x é um número inteiro positivo?
10. Encontre um fator primo de 10 números inteiros ímpares diferentes, cada um com 20 dígitos, selecionados aleatoriamente. Veja quanto tempo leva para encontrar um fator de cada um desses números inteiros. Faça a mesma coisa para 10 números inteiros ímpares de 30 dígitos e para 10 números inteiros ímpares de 40 dígitos e assim por diante, continuando até onde for possível.
11. Encontre todos os pseudoprimos na base 2, ou seja, números inteiros compostos n tal que $2^{n-1} \equiv 1 \pmod{n}$, em que n não excede 10 000.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Examine a história da palavra *algoritmo* e descreva o seu uso nos primeiros escritos.
2. Veja a introdução original de Bachmann sobre a notação O grande. Explique como ele e outros usavam essa notação.
3. Explique como os algoritmos de ordenação podem ser classificados em uma taxonomia baseada no princípio fundamental no qual eles são baseados.
4. Descreva o algoritmo radix sort.
5. Descreva qual o significado de um algoritmo paralelo. Explique como o pseudocódigo usado neste livro pode ser estendido aos algoritmos paralelos.
6. Explique como a complexidade dos algoritmos paralelos pode ser medida. Dê alguns exemplos para ilustrar esse conceito, mostrando como um algoritmo paralelo pode trabalhar mais rapidamente do que aquele que não opera em paralelo.
7. Descreva o teste de Lucas-Lehmer para determinar se um número de Mersenne é primo. Discuta o progresso do projeto GIMPS em encontrar números primos de Mersenne usando esse teste.
8. Explique como os testes de probabilidade de primalidade são usados na prática para produzir números extremamente grandes que são certamente quase primos. Tais testes têm alguma desvantagem em potencial?
9. A questão sobre se há um número infinito de números de Carmichael foi resolvida recentemente, depois de estar em aberto por mais de 75 anos. Descreva os elementos que foram usados na demonstração de que eles são em número infinito.
10. Resuma o status atual dos algoritmos de fatoração em termos de sua complexidade e do tamanho dos números que podem ser fatorados. Quando você acha que será possível fatorar um número com 200 dígitos?
11. Descreva os algoritmos que são usados atualmente em computadores modernos para adicionar, subtrair, multiplicar e dividir números inteiros positivos.
12. Descreva a história do Teorema Chinês do Resto. Fale sobre alguns problemas relevantes abordados nos escritos chineses e hindus e como o Teorema do Resto se aplica a eles.
13. Quando os números de uma seqüência são realmente aleatórios e quando não são? Quais as desvantagens observadas em simulações e experimentos nos quais os números pseudo-aleatórios são usados? Quais são as propriedades que os números pseudo-aleatórios podem ter que os números aleatórios não têm?
14. Descreva como a criptografia de chave pública é aplicada. As maneiras como elas são aplicadas são seguras, dado o status dos algoritmos de fatoração? A informação que é mantida em segredo com a criptografia de chave pública pode deixar de ser segura no futuro?
15. Descreva como a criptografia de chave pública pode ser usada para mandar mensagens secretas assinadas para que o destinatário tenha certeza de que ela foi enviada por aquele que diz ter mandado.
16. Mostre como uma congruência pode ser usada para dizer o dia da semana de qualquer data especificada.
17. Descreva alguns algoritmos usados para multiplicar de forma eficiente números inteiros grandes.
18. Descreva alguns algoritmos usados para multiplicar matrizes grandes de forma eficiente.
19. Descreva o criptossistema de chave pública Rabin, explicando como codificar e como decodificar mensagens e por que é razoável usá-lo como um criptossistema de chave pública.

CAPÍTULO

4 Indução e Recursão

4.1 Indução Matemática
4.2 Indução Completa e Boa Ordenação
4.3 Definições Recursivas e Indução Estrutural
4.4 Algoritmos Recursivos
4.5 Exatidão de Programas

Muitas proposições matemáticas afirmam que uma propriedade é verdadeira para todos os números inteiros positivos. Alguns exemplos dessas proposições são: para todo número inteiro positivo n: $n! \leq n^n$, $n^3 - n$ é divisível por 3; um conjunto com n elementos tem 2^n subconjuntos e a soma dos primeiros n números inteiros positivos é $n(n+1)/2$. Um dos objetivos principais deste capítulo, e do livro, é dar ao estudante um entendimento da indução matemática, que é utilizada para demonstrar resultados desse tipo.

As demonstrações com o uso da indução matemática têm duas partes. Primeiro, elas mostram que a proposição é verdadeira para o número inteiro positivo 1. Segundo, elas mostram que se a proposição for verdadeira para um número inteiro positivo, então deve ser mantida para o número inteiro seguinte. A indução matemática baseia-se na regra de inferência, que nos diz que se $P(1)$ e $\forall k(P(k) \rightarrow P(k+1))$ são verdadeiras para o domínio dos números inteiros positivos, então $\forall P(n)$ é verdadeira. A indução matemática pode ser usada para demonstrar diversos resultados. Entender como ler e construir demonstrações por indução matemática é a chave para aprender matemática discreta.

Nos capítulos 2 e 3, definimos conjuntos e funções, ou seja, descrevemos os conjuntos listando seus elementos ou dando propriedades que caracterizassem seus elementos. Fornecemos fórmulas para os valores das funções. Há outra maneira importante de definir esses objetos, baseando-se em indução matemática. Para definir as funções, alguns termos iniciais são especificados e uma regra é dada para encontrar os valores subseqüentes a partir daqueles já conhecidos. Os conjuntos podem ser definidos listando-se alguns de seus elementos e dando-se regras para a construção de elementos a partir daqueles já conhecidos como pertencendo ao conjunto. Essas definições, chamadas de *definições recursivas*, são usadas pela matemática discreta e pela ciência da computação. Uma vez que definimos um conjunto recursivamente, podemos usar um método de demonstração chamado de indução estrutural, ou recursão, para mostrar os resultados sobre esse conjunto.

Quando um procedimento é específico para a resolução de um problema, ele deve *sempre* resolver o problema corretamente. Apenas testar para ver se o resultado obtido para um conjunto com valores de entrada é correto não mostra se o procedimento trabalha sempre corretamente. A exatidão de um procedimento apenas pode ser garantida demonstrando-se que ele sempre fornece o resultado correto. A seção final deste capítulo contém uma introdução sobre as técnicas de verificação de programas. Esta é uma técnica formal para verificar se um procedimento está correto. A verificação de programas serve como base para a tentativa de demonstrar de modo mecânico que os programas estão corretos.

4.1 Indução Matemática

Introdução

Suponha que tenhamos uma escada infinita, como mostrada na Figura 1, e queremos saber se podemos alcançar todos os degraus dessa escada. Sabemos duas coisas:

1. Podemos alcançar o primeiro degrau da escada.
2. Se pudermos alcançar um determinado degrau da escada, então poderemos alcançar o próximo degrau.

Podemos concluir que nós podemos alcançar todos os degraus? Por (1), sabemos que podemos alcançar o primeiro degrau da escada. Além disso, como podemos alcançar o primeiro, por (2), podemos também alcançar o segundo degrau, que é o próximo degrau depois do primeiro. Aplicando (2) novamente, como podemos alcançar o segundo degrau, podemos também alcançar o terceiro. Continuando dessa maneira, podemos mostrar que será possível alcançar o quarto degrau, o quinto degrau, e assim por diante. Por exemplo, depois de usar 100 vezes (2), sabemos

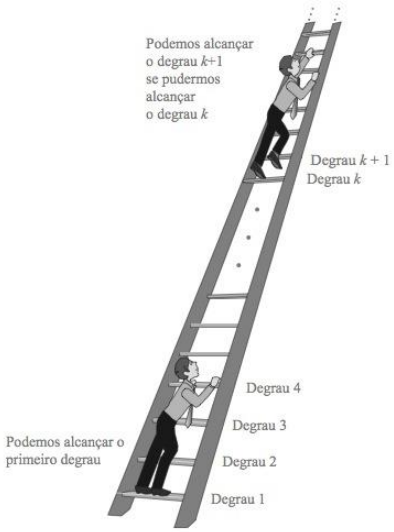


FIGURA 1 Subindo uma Escada Infinita.

que poderemos alcançar o 101º degrau. Mas podemos concluir que é possível alcançar todos os degraus dessa escada infinita? A resposta é sim, às vezes podemos fazer essa verificação usando uma importante técnica de demonstração chamada de **indução matemática**. Ou seja, podemos mostrar que $P(n)$ é verdadeira para todo número inteiro positivo n, em que $P(n)$ é a proposição que afirma que podemos alcançar o n-ésimo degrau da escada.

A indução matemática é uma técnica de demonstração extremamente importante que pode utilizada para apresentar declarações desse tipo. Como veremos nesta seção e nas subseqüentes, a indução matemática é muito usada para demonstrar resultados sobre diversos objetos discretos. Por exemplo, ela é aplicada para demonstrar resultados sobre complexidade de algoritmos, exatidão de certos tipos de programas, teoremas sobre grafos e árvores, assim como várias identidades e inequações.

Nesta seção, descreveremos como a indução matemática pode ser usada e por que é uma técnica de demonstração válida. É extremamente importante notar que a indução matemática pode ser utilizada apenas para demonstrar resultados obtidos de outras formas. Ela *não* é um instrumento para descobrir fórmulas ou teoremas.

Indução Matemática

Auto-avaliação

Em geral, a indução matemática* pode ser usada para demonstrar proposições que afirmam que $P(n)$ é verdadeira para todos os números inteiros positivos n, em que $P(n)$ é uma função proposicional. Uma demonstração por indução matemática tem duas partes, um **passo base**, em que mostramos que $P(1)$ é verdadeira, e um **passo de indução**, em que mostramos que para todos os números inteiros k, se $P(k)$ for verdadeira, então $P(k + 1)$ é verdadeira.

* Infelizmente, usar a terminologia "indução matemática" entra em confronto com a terminologia usada para descrever diferentes tipos de argumentos. Em lógica, o **argumento dedutivo** usa regras de inferência para construir as conclusões a partir de premissas, enquanto o **argumento indutivo** obtém conclusões apenas apoiado, não assegurado, em evidências. As demonstrações matemáticas, que incluem argumentos que usam a indução matemática, são dedutivas, não indutivas.

PRINCÍPIO DA INDUÇÃO MATEMÁTICA Para demonstrar que $P(n)$ é verdadeira para todos os números inteiros n, em que $P(n)$ é uma função proposicional, completamos dois passos:

PASSO BASE: Verificamos que $P(1)$ é verdadeira.

PASSO DE INDUÇÃO: Mostramos que a proposição condicional $P(k) \rightarrow P(k+1)$ é verdadeira para todos os números inteiros positivos k.

Para completar o passo de indução de uma demonstração que utiliza o princípio da indução matemática, assumimos que $P(k)$ seja verdadeira para um número inteiro positivo k arbitrário e mostramos, sob hipótese, que $P(k+1)$ deve ser também verdadeira. A hipótese de que $P(k)$ é verdadeira é chamada de **hipótese indutiva**. Uma vez completados os dois passos da demonstração por indução matemática, mostramos que $P(n)$ é verdadeira para todos os números inteiros positivos, ou seja, evidenciamos que $\forall n\, P(n)$ é verdadeira onde esta quantificação tem como domínio o conjunto dos números inteiros positivos. No passo de indução, mostramos que $\forall k(P(k) \rightarrow P(k+1))$ é verdadeira onde novamente o domínio é o conjunto dos números inteiros positivos.

Expressa como uma regra de inferência, essa técnica de demonstração pode ser declarada como

$$[P(1) \wedge \forall\, k(P(k) \rightarrow P(k+1))] \rightarrow \forall\, n\, P(n),$$

em que o domínio é o conjunto dos números inteiros positivos. Como a indução matemática é uma importante técnica, é válido explicar em detalhes os passos de uma demonstração que usa essa técnica. A primeira coisa que fazemos ao demonstrar que $P(n)$ é verdadeira para todos os números inteiros positivos n é mostrar que $P(1)$ é verdadeira. Ou seja, devemos evidenciar que a proposição obtida quando n é substituído por 1 em $P(n)$ é verdadeira. Então, devemos evidenciar que $P(k) \rightarrow P(k+1)$ é verdadeira para todos os números inteiros positivos k. Para demonstrar que essa proposição condicional é verdadeira para todo número inteiro positivo k, precisamos evidenciar que $P(k+1)$ não pode ser falsa quando $P(k)$ é verdadeira. Isso pode ser feito assumindo-se que $P(k)$ é verdadeira e mostrando que, *a partir dessa hipótese*, $P(k+1)$ deve também ser verdadeira.

Lembre-se: Em uma demonstração por indução matemática *não* assumimos que $P(k)$ seja verdadeira para todos os números inteiros positivos! Apenas mostramos que *se assumimos* que $P(k)$ é verdadeira, então $P(k+1)$ também é verdadeira. Assim, uma demonstração por indução matemática não pode ser vista como um caso de usar o que se quer demonstrar, ou um argumento circular.

Quando usamos a indução matemática para demonstrar um teorema, primeiro mostramos que $P(1)$ é verdadeira. Então, sabemos que $P(2)$ é verdadeira, porque $P(1)$ implica $P(2)$. Assim, sabemos que $P(3)$ é verdadeira, porque $P(2)$ implica $P(3)$. Continuando nessa linha, vemos que $P(n)$ é verdadeira para todo número inteiro positivo n.

MANEIRAS DE LEMBRAR COMO A INDUÇÃO MATEMÁTICA FUNCIONA Pensar em uma escada infinita e nas regras para alcançar os degraus pode ajudá-lo a lembrar como a indução matemática funciona. Note que as proposições (1) e (2) para a escada infinita são exatamente o passo base e o de indução, respectivamente, da demonstração de que $P(n)$ é verdadeira para todos os números inteiros positivos n, em que $P(n)$ é a proposição que afirma que podemos alcançar o n-ésimo degrau da escada. Conseqüentemente, podemos invocar a indução matemática para concluir que podemos alcançar todos os degraus.

Links

NOTA HISTÓRICA O primeiro uso conhecido da indução matemática é o trabalho do matemático Francesco Maurolico (1494–1575), no século XVI. Maurolico escreveu exaustivamente sobre os trabalhos de matemáticos clássicos e fez muitas contribuições em geometria e ótica. Em seu livro, *Arithmeticorum Libri Duo,* Maurolico apresentou várias propriedades de números inteiros juntamente com as demonstrações dessas propriedades. Para demonstrar algumas dessas propriedades, ele desenvolveu o método de indução matemática. Seu primeiro uso da indução matemática em seu livro foi para demonstrar que a soma dos primeiros n números inteiros positivos e ímpares é igual a n^2.

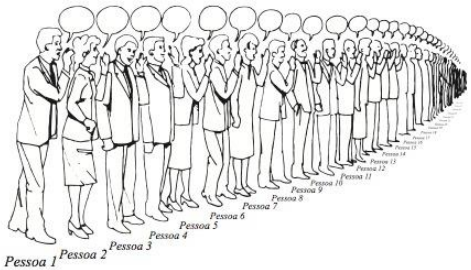


FIGURA 2 Pessoas Contando Segredos.

Além da escada infinita, muitas outras ilustrações úteis da indução matemática podem ajudá-lo a lembrar como esse princípio funciona. Uma delas contém uma fila de pessoas, a pessoa um, a pessoa dois, e assim por diante. Um segredo é contado à pessoa um e cada pessoa conta o segredo à próxima da fila, se a pessoa anterior ouvi-lo. Considere $P(n)$ como a proposição: a pessoa n sabe o segredo. Então $P(1)$ é verdadeira, porque o segredo é contado à pessoa um; $P(2)$ é verdadeira, porque a pessoa um conta o segredo à pessoa dois; $P(3)$ é verdadeira, porque a pessoa dois conta o segredo à terceira pessoa, e assim por diante. Pelo princípio da indução matemática, cada pessoa da fila ouve o segredo. Isso é ilustrado na Figura 2. (Assumimos que cada pessoa repassa o segredo, sem mudá-lo, para a próxima pessoa, o que geralmente não é verdade na vida real.)

Outra maneira de ilustrar o princípio da indução matemática é considerar uma fila infinita de dominós, marcados a partir de 1, 2, 3, . . . , n, . . . , em que cada dominó está em pé. Considere $P(n)$ como a proposição: o dominó n caiu. Se o primeiro dominó cair, considerando-se que $P(1)$ seja verdadeira, e se, sempre que o k-ésimo dominó cair, o $(k + 1)$-ésimo dominó também vai cair, ou seja, se $P(k) \rightarrow P(k + 1)$ for verdadeira para todo número inteiro positivo k —, então todos os dominós vão cair. Isso é ilustrado na Figura 3.

Exemplos de Demonstrações por Indução Matemática

Muitos teoremas afirmam que $P(n)$ é verdadeira para todos os números inteiros positivos n, em que $P(n)$ é uma função proposicional, tal como a proposição de que $1 + 2 + \cdots + n = n(n + 1)/2$ para todo número inteiro positivo n ou a proposição $n \leq 2^n$ para todos os números inteiros positivos n.

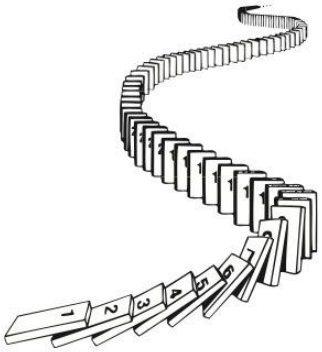


FIGURA 3 Ilustrando a Indução Matemática com o Uso de Dominós.

A indução matemática é uma técnica para demonstrar teoremas desse tipo. Em outras palavras, ela pode ser utilizada para demonstrar proposições na forma $\forall n\, P(n)$, em que o domínio é o conjunto dos números inteiros positivos. Pode ser usada também para demonstrar diversos teoremas, sendo cada um deles uma proposição nessa forma.

Links

Usaremos vários exemplos para ilustrar como os teoremas são demonstrados aplicando-se a indução matemática. Os teoremas que vamos demonstrar incluem fórmulas de somatório, inequações, identidades para combinações de conjuntos, resultados de divisibilidade, teoremas sobre algoritmos e alguns outros resultados criativos. Nas seções posteriores, vamos empregar a indução matemática para demonstrar muitos outros tipos de resultados, incluindo exatidão de programas de computação e algoritmos. Esta pode ser utilizada para demonstrar vários teoremas, não apenas fórmulas de somatório, inequações e outros tipos de exemplos que ilustramos aqui. Note que há muitas oportunidades para erro nas demonstrações por indução. Vamos descrever algumas demonstrações incorretas por indução matemática no final desta seção e nos exercícios.

DEMONSTRAÇÃO DE FÓRMULAS DE SOMATÓRIO Comecemos usando a indução matemática para demonstrar muitas fórmulas de somatório diferentes. Como veremos, ela é particularmente apropriada para demonstrar que essas fórmulas são válidas. Entretanto, as fórmulas de somatório podem ser demonstradas de outras maneiras. Isso não é surpreendente porque existem normalmente maneiras diferentes para demonstrar um teorema. A maior desvantagem do uso da indução matemática é que ela não pode ser utilizada para encontrar uma fórmula de somatório, ou seja, é necessário ter a fórmula antes de começar a prová-la com a indução matemática. Começamos utilizando-a para demonstrar uma fórmula para a soma dos n menores números inteiros positivos.

EXEMPLO 1 Mostre que se n for um número inteiro positivo, então

Exemplos Extras

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

Solução: Considere $P(n)$ como a proposição que afirma que a soma dos primeiros n números inteiros positivos é $n(n+1)/2$. Devemos fazer duas coisas para demonstrar que $P(n)$ é verdadeira para $n = 1, 2, 3, \ldots$. Ou seja, precisamos mostrar que $P(1)$ é verdadeira e que a proposição condicional $P(k)$ implica que $P(k+1)$ é verdadeira para $k = 1, 2, 3, \ldots$.

PASSO BASE: $P(1)$ é verdadeira, pois $1 = \frac{1(1+1)}{2}$.

PASSO DE INDUÇÃO: Para a hipótese indutiva, assumimos que $P(k)$ é verdadeira para um número inteiro positivo arbitrário k, ou seja, assumimos que

$$1 + 2 + \cdots + k = \frac{k(k+1)}{2}.$$

Considerando essa hipótese, devemos mostrar que $P(k+1)$ é verdadeira, ou seja, que

$$1 + 2 + \cdots + k + (k+1) = \frac{(k+1)[(k+1)+1]}{2} = \frac{(k+1)(k+2)}{2}$$

também é verdadeira. Quando adicionamos $k+1$ nos dois lados da equação em $P(k)$, obtemos

$$\begin{aligned} 1 + 2 + \cdots + k + (k+1) &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2}. \end{aligned}$$

Esta última equação mostra que $P(k+1)$ é verdadeira, considerando a hipótese de que $P(k)$ seja verdadeira. Isso completa o passo de indução.

Completamos os passos base e de indução. Assim, pela indução matemática, sabemos que $P(n)$ é verdadeira para todos os números inteiros positivos n, ou seja, demonstramos que $1 + 2 + \cdots + n = n(n+1)/2$ para todos os números inteiros positivos n. ◀

Como notamos, a indução matemática não é um instrumento para encontrar teoremas sobre todos os números inteiros positivos. Em vez disso, é um método de demonstração de resultados, uma vez conjecturados. No Exemplo 2, usando a indução matemática para demonstrar uma fórmula de somatório, vamos formular e então demonstrar uma conjectura.

EXEMPLO 2 Conjecture uma fórmula para a soma dos primeiros n números inteiros positivos e ímpares. Então, demonstre sua conjectura usando a indução matemática.

Solução: As somas dos primeiros n números inteiros positivos e ímpares para $n = 1, 2, 3, 4, 5$ são

$$1 = 1, \qquad 1 + 3 = 4, \qquad 1 + 3 + 5 = 9,$$
$$1 + 3 + 5 + 7 = 16, \qquad 1 + 3 + 5 + 7 + 9 = 25.$$

A partir desses valores, é razoável conjecturar que a soma dos primeiros n números inteiros positivos e ímpares é n^2, ou seja, $1 + 3 + 5 + \cdots + (2n-1) = n^2$. Precisamos de um método para *demonstrar* que esta *conjectura* está correta, se de fato ela estiver.

Considere $P(n)$ como a proposição: a soma dos primeiros n números inteiros positivos e ímpares é n^2. Nossa conjectura é que $P(n)$ seja verdadeira para todos os números inteiros positivos. Para usar a indução matemática com o intuito de demonstrar essa conjectura, devemos primeiro completar o passo base; ou seja, devemos mostrar que $P(1)$ é verdadeira. Então, iremos para o passo de indução; ou seja, devemos mostrar que $P(k+1)$ é verdadeira quando assumimos que $P(k)$ seja verdadeira. Tentaremos agora completar esses dois passos.

PASSO BASE: $P(1)$ declara que a soma do primeiro número inteiro positivo ímpar é 1^2. Isso é verdadeiro porque a soma do primeiro número inteiro positivo e ímpar é 1. O passo base está completo.

PASSO DE INDUÇÃO: Para completar o passo de indução, devemos mostrar que a proposição $P(k) \rightarrow P(k+1)$ é verdadeira para todo número inteiro positivo k. Para fazer isso, primeiro assumimos a hipótese indutiva. A hipótese indutiva é a proposição: $P(k)$ é verdadeira, ou seja,

$$1 + 3 + 5 + \cdots + (2k-1) = k^2.$$

[Note que o k-ésimo número inteiro positivo e ímpar é $(2k-1)$, pois este inteiro é obtido pela adição de 2 em um total de $k-1$ vezes 1.] Para mostrar que $\forall k\,(P(k) \rightarrow P(k+1))$ é verdadeira, devemos mostrar que se $P(k)$ for verdadeira (a hipótese indutiva), então, $P(k+1)$ é verdadeira. Note que $P(k+1)$ é a proposição de que

$$1 + 3 + 5 + \cdots + (2k-1) + (2k+1) = (k+1)^2.$$

Então, assumindo que $P(k)$ seja verdadeira, temos que

$$\begin{aligned} 1 + 3 + 5 + \cdots + (2k-1) + (2k+1) &= [1 + 3 + \cdots + (2k-1)] + (2k+1) \\ &= k^2 + (2k+1) \\ &= k^2 + 2k + 1 \\ &= (k+1)^2. \end{aligned}$$

Isso mostra que $P(k+1)$ resulta de $P(k)$. Note que usamos a hipótese indutiva $P(k)$ na segunda equação para substituir a soma dos primeiros k números inteiros positivos e ímpares por k^2.

Completamos o passo base e o de indução, ou seja, mostramos que $P(1)$ é verdadeira e a proposição condicional $P(k) \rightarrow P(k+1)$ é verdadeira para todos os números inteiros positivos k. Conseqüentemente, pelo princípio da indução matemática, podemos concluir que $P(n)$ é verdadeira para todos os números inteiros positivos n. Ou seja, sabemos que $1 + 3 + 5 + \cdots + (2n-1) = n^2$ para todos os números inteiros positivos n. ◀

Às vezes, precisamos mostrar que $P(n)$ é verdadeira para $n = b, b+1, b+2, \ldots$, em que b é um número inteiro diferente de 1. Podemos usar a indução matemática para fazer isso mudando o passo base. Por exemplo, considere o Exemplo 3, que afirma que a fórmula do somatório é válida para todos os números inteiros não negativos, assim, precisamos demonstrar que $P(n)$ é verdadeira para $n = 0, 1, 2, \ldots$.

EXEMPLO 3 Use a indução matemática para mostrar que

$$1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$$

para todos os números inteiros não negativos n.

Solução: Considere $P(n)$ como a proposição de que $1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$ para o número inteiro n.

PASSO BASE: $P(0)$ é verdadeira, porque $2^0 = 1 = 2^1 - 1$. Isso completa a etapa de base.

PASSO DE INDUÇÃO: Para a hipótese indutiva, assumimos que $P(k)$ é verdadeira, ou seja, assumimos que

$$1 + 2 + 2^2 + \cdots + 2^k = 2^{k+1} - 1.$$

Para realizar o passo de indução usando essa hipótese, devemos mostrar que quando assumimos que $P(k)$ é verdadeira, então $P(k+1)$ é também verdadeira, ou seja, devemos mostrar que

$$1 + 2 + 2^2 + \cdots + 2^k + 2^{k+1} = 2^{(k+1)+1} - 1 = 2^{k+2} - 1$$

assumindo a hipótese indutiva $P(k)$. Considerando-se essa hipótese de $P(k)$, vemos que

$$\begin{aligned} 1 + 2 + 2^2 + \cdots + 2^k + 2^{k+1} &= (1 + 2 + 2^2 + \cdots + 2^k) + 2^{k+1} \\ &= (2^{k+1} - 1) + 2^{k+1} \\ &= 2 \cdot 2^{k+1} - 1 \\ &= 2^{k+2} - 1. \end{aligned}$$

Note que usamos a hipótese indutiva na segunda equação nessa seqüência de equações para substituir $1 + 2 + 2^2 + \cdots + 2^k$ por $2^{k+1} - 1$. Completamos o passo de indução.

Como completamos o passo base e o de indução pela indução matemática, sabemos que $P(n)$ é verdadeira para todos os números inteiros não negativos n, ou seja, $1 + 2 + \cdots + 2^n = 2^{n+1} - 1$ para todos os números inteiros não negativos n. ◀

No Exemplo 3, no passo base, mostramos que $P(0)$ é verdadeira, porque o teorema que queríamos demonstrar estava na forma $\forall n\, P(n)$, em que o domínio era o conjunto de todos os números inteiros não negativos. Como esse exemplo demonstra, para usar a indução matemática para mostrar que $P(n)$ é verdadeira para todo $n = b, b+1, b+2, \ldots$, em que b é um número inteiro diferente de 1, mostramos que $P(b)$ é verdadeira (no passo base) e, então, mostramos que a proposição condicional $P(k) \rightarrow P(k+1)$ é verdadeira para $k = b, b+1, b+2, \ldots$ (no passo de indução). Note que b pode ser negativo, zero ou positivo. Seguindo com a analogia do dominó que usamos anteriormente, imagine que começamos derrubando o b-ésimo dominó (no passo

base), e quando cada um dos dominós cai, empurra o próximo dominó (no passo de indução). Deixamos para o leitor mostrar que essa forma de indução é válida (veja o Exercício 79).

A fórmula dada no Exemplo 3 é um caso especial de resultado geral para a soma dos termos de uma progressão geométrica (Teorema 1 da Seção 2.4). Usaremos a indução matemática para fornecer uma demonstração alternativa dessa fórmula.

EXEMPLO 4 **Somas de Progressões Geométricas** Use a indução matemática para demonstrar essa fórmula para a soma de um número finito de termos de uma progressão geométrica:

$$\sum_{j=0}^{n} ar^j = a + ar + ar^2 + \cdots + ar^n = \frac{ar^{n+1} - a}{r - 1} \qquad \text{quando } r \neq 1,$$

em que n é um número inteiro não negativo.

Solução: Para demonstrar essa fórmula usando a indução matemática, considere $P(n)$ como a proposição: a soma dos primeiros $n + 1$ termos de uma progressão geométrica nesta fórmula está correta.

PASSO BASE: $P(0)$ é verdadeira, porque

$$\frac{ar^{0+1} - a}{r - 1} = \frac{ar - a}{r - 1} = \frac{a(r - 1)}{r - 1} = a.$$

PASSO DE INDUÇÃO: A hipótese indutiva é que a proposição $P(k)$ seja verdadeira, em que k é um número inteiro não negativo, ou seja, $P(k)$ é a proposição que

$$a + ar + ar^2 + \cdots + ar^k = \frac{ar^{k+1} - a}{r - 1}.$$

Para completar o passo de indução, devemos mostrar que se $P(k)$ for verdadeira, então $P(k + 1)$ é também verdadeira. Para mostrarmos que este é o caso, primeiro adicionamos ar^{k+1} aos dois lados da equação declarada por $P(k)$. Verificamos que

$$a + ar + ar^2 + \cdots + ar^k + ar^{k+1} = \frac{ar^{k+1} - a}{r - 1} + ar^{k+1}.$$

Reescrevendo o lado direito dessa equação, temos que

$$\begin{aligned} \frac{ar^{k+1} - a}{r - 1} + ar^{k+1} &= \frac{ar^{k+1} - a}{r - 1} + \frac{ar^{k+2} - ar^{k+1}}{r - 1} \\ &= \frac{ar^{k+2} - a}{r - 1}. \end{aligned}$$

Combinando essas duas últimas equações, temos

$$a + ar + ar^2 + \cdots + ar^k + ar^{k+1} = \frac{ar^{k+2} - a}{r - 1}.$$

Isso mostra que se a hipótese indutiva $P(k)$ for verdadeira, então $P(k + 1)$ deverá também ser verdadeira. Isso completa o argumento indutivo.

Completamos o passo base e o de indução; então, pela indução matemática, $P(n)$ é verdadeira para todos os números inteiros não negativos n. Isso mostra que a fórmula para a soma dos termos de uma progressão geométrica está correta. ◀

Como mencionado anteriormente, a fórmula do Exemplo 3 é o caso da fórmula do Exemplo 4 com $a = 1$ e $r = 2$. O leitor deve verificar que colocar esses valores para a e r na fórmula geral dá a mesma fórmula que a do Exemplo 3.

DEMONSTRANDO INEQUAÇÕES A indução matemática pode ser usada para demonstrar que diversas inequações são verdadeiras para todos os números inteiros positivos maiores que um determinado número inteiro positivo, como ilustram os exemplos 5 a 7.

EXEMPLO 5 Use a indução matemática para demonstrar a inequação

Exemplos Extras

$$n < 2^n$$

para todos os números inteiros positivos n.

Solução: Considere $P(n)$ como a proposição $n < 2^n$.

PASSO BASE: $P(1)$ é verdadeira, porque $1 < 2^1 = 2$. Isso completa o passo base.

PASSO DE INDUÇÃO: Primeiro, assumimos a hipótese indutiva de que $P(k)$ é verdadeira para o número inteiro positivo k, ou seja, a hipótese indutiva $P(k)$ é a proposição de que $k < 2^k$. Para completar o passo de indução, precisamos mostrar que, se $P(k)$ for verdadeira, então $P(k + 1)$, que é a proposição de que $k + 1 < 2^{k+1}$, é verdadeira. Ou seja, precisamos mostrar que se $k < 2^k$, então $k + 1 < 2^{k+1}$. Para mostrar que esta proposição condicional é verdadeira para o número inteiro positivo k, primeiro adicionamos 1 aos dois lados de $k < 2^k$, e então notamos que $1 \leq 2^k$. Isso nos diz que

$$k + 1 < 2^k + 1 \leq 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}.$$

Isso mostra que $P(k + 1)$ é verdadeira, ou seja, que $k + 1 < 2^{k+1}$, com base na hipótese de que $P(k)$ seja verdadeira. O passo de indução está completo.

Assim, como completamos o passo base e o de indução, pelo princípio da indução matemática mostramos que $n < 2^n$ é verdadeira para todos os números inteiros positivos n. ◀

EXEMPLO 6 Use a indução matemática para demonstrar que $2^n < n!$ para todo número inteiro positivo n com $n \geq 4$. (Note que essa inequação é falsa para $n = 1, 2$ e 3.)

Solução: Considere $P(n)$ como a proposição: $2^n < n!$.

PASSO BASE: Para demonstrar a inequação para $n \geq 4$ é necessário que o passo base seja $P(4)$. Note que $P(4)$ é verdadeira, porque $2^4 = 16 < 24 = 4!$

PASSO DE INDUÇÃO: Para este passo, assumimos que $P(k)$ seja verdadeira para o número inteiro positivo k com $k \geq 4$. Ou seja, assumimos que $2^k < k!$ para o número inteiro positivo k com $k \geq 4$. Devemos mostrar que, considerando-se essa hipótese, $P(k + 1)$ também é verdadeira, ou seja, devemos mostrar que se $2^k < k!$ para o número inteiro positivo k, em que $k \geq 4$, então $2^{k+1} < (k + 1)!$. Temos que

$$\begin{aligned} 2^{k+1} &= 2 \cdot 2^k && \text{pela definição de expoente} \\ &< 2 \cdot k! && \text{pela hipótese indutiva} \\ &< (k + 1)k! && \text{pois } 2 < k + 1 \\ &= (k + 1)! && \text{pela definição de função fatorial.} \end{aligned}$$

Isso mostra que $P(k + 1)$ é verdadeira quando $P(k)$ é verdadeira, o que completa o passo de indução da demonstração.

Completamos o passo base e o de indução. Assim, pela indução matemática, $P(n)$ é verdadeira para todos os números inteiros n com $n \geq 4$, ou seja, demonstramos que $2^n < n!$ é verdadeira para todos os números inteiros n com $n \geq 4$. ◀

Uma inequação importante para a soma dos recíprocos de um conjunto de números inteiros positivos será demonstrada no Exemplo 7.

EXEMPLO 7 **Uma Inequação para os Números Harmônicos** Os **números harmônicos** H_j, $j = 1, 2, 3, \ldots$, são definidos por

$$H_j = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{j}.$$

Por exemplo,

$$H_4 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{25}{12}.$$

Use a indução matemática para mostrar que

$$H_{2^n} \geq 1 + \frac{n}{2},$$

sempre que n for um número inteiro não negativo.

Solução: Para realizar a demonstração, considere $P(n)$ como a proposição de que $H_{2^n} \geq 1 + \frac{n}{2}$.

PASSO BASE: $P(0)$ é verdadeira, porque $H_{2^0} = H_1 = 1 \geq 1 + \frac{0}{2}$.

PASSO DE INDUÇÃO: A hipótese indutiva é a proposição de que $P(k)$ seja verdadeira, ou seja, $H_{2^k} \geq 1 + \frac{k}{2}$, em que k é um número inteiro não negativo. Devemos mostrar que, se $P(k)$ for verdadeira, então $P(k + 1)$, que afirma que $H_{2^{k+1}} \geq 1 + \frac{k+1}{2}$, também é verdadeira. Então, assumindo a hipótese indutiva, temos que

$$\begin{aligned}
H_{2^{k+1}} &= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{2^k} + \frac{1}{2^k + 1} + \cdots + \frac{1}{2^{k+1}} && \text{pela definição de número harmônico} \\
&= H_{2^k} + \frac{1}{2^k + 1} + \cdots + \frac{1}{2^{k+1}} && \text{pela definição do } 2^k\text{-ésimo número harmônico} \\
&\geq \left(1 + \frac{k}{2}\right) + \frac{1}{2^k + 1} + \cdots + \frac{1}{2^{k+1}} && \text{pela hipótese indutiva} \\
&\geq \left(1 + \frac{k}{2}\right) + 2^k \cdot \frac{1}{2^{k+1}} && \text{porque há } 2^k \text{ termos, cada um} \geq 1/2^{k+1} \\
&\geq \left(1 + \frac{k}{2}\right) + \frac{1}{2} && \text{cancelando um fator comum de } 2^k \text{ no segundo termo} \\
&= 1 + \frac{k+1}{2}.
\end{aligned}$$

Isso conclui o passo de indução da demonstração.

Completamos o passo base e o de indução. Assim, pela indução matemática, $P(n)$ é verdadeira para todos os números inteiros não negativos, ou seja, a inequação $H_{2^n} \geq 1 + \frac{n}{2}$ para números harmônicos é válida para todos os números inteiros não negativos n. ◀

Lembre-se: A inequação estabelecida aqui mostra que a **série harmônica**

$$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} + \cdots$$

é uma série infinita divergente. Este é um importante exemplo no estudo de séries infinitas.

DEMONSTRANDO RESULTADOS DE DIVISIBILIDADE A indução matemática pode ser utilizada para demonstrar resultados de divisibilidade de números inteiros. Embora esses resultados sejam geralmente fáceis de ser demonstrados por meio de resultados básicos em teoria dos números, é interessante ver como demonstrar resultados usando indução matemática, como ilustra o Exemplo 8.

EXEMPLO 8 Use a indução matemática para demonstrar que $n^3 - n$ é divisível por 3 sempre que n for um número inteiro positivo.

Exemplos Extras

Solução: Para construir a demonstração, considere $P(n)$ como a proposição: "$n^3 - n$ é divisível por 3."

PASSO BASE: A proposição $P(1)$ é verdadeira porque $1^3 - 1 = 0$ é divisível por 3. Isso completa o passo base.

PASSO DE INDUÇÃO: Para a hipótese indutiva, assumimos que $P(k)$ seja verdadeira; ou seja, assumimos que $k^3 - k$ seja divisível por 3. Para completar este passo, devemos mostrar que quando assumimos a hipótese indutiva, temos que $P(k + 1)$, ou seja, a proposição: $(k + 1)^3 - (k + 1)$ é divisível por 3, é também verdadeira. Logo, devemos mostrar que $(k + 1)^3 - (k + 1)$ é divisível por 3. Note que

$$\begin{aligned}(k+1)^3 - (k+1) &= (k^3 + 3k^2 + 3k + 1) - (k+1) \\ &= (k^3 - k) + 3(k^2 + k).\end{aligned}$$

Como ambos os termos dessa soma são divisíveis por 3 (o primeiro pela hipótese indutiva e o segundo porque é 3 vezes um número inteiro), temos que $(k + 1)^3 - (k + 1)$ é também divisível por 3. Isso completa o passo de indução.

Como nós completamos o passo base e o de indução, pelo princípio de indução matemática sabemos que $n^3 - n$ é divisível por 3 sempre que n for um número inteiro positivo. ◀

DEMONSTRANDO RESULTADOS DE CONJUNTOS A indução matemática pode ser utilizada para demonstrar muitos resultados de conjuntos. Em particular, no Exemplo 9 demonstramos uma fórmula para o número de subconjuntos de um conjunto finito, e no Exemplo 10 estabelecemos uma identidade de conjuntos.

EXEMPLO 9 **O Número de Subconjuntos de um Conjunto Finito** Use a indução matemática para mostrar que se S for um conjunto finito com n elementos em que n é um número inteiro não negativo, então S tem 2^n subconjuntos. (Vamos demonstrar esse resultado diretamente de muitas maneiras no Capítulo 5.)

Solução: Considere $P(n)$ como a proposição de que um conjunto com n elementos tem 2^n subconjuntos.

PASSO BASE: $P(0)$ é verdadeira, porque um conjunto com zero elementos, o conjunto vazio, tem exatamente $2^0 = 1$ subconjunto, ou seja, ele mesmo.

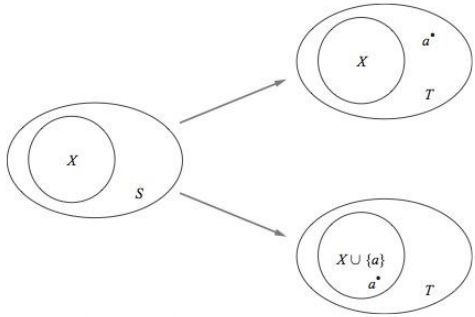


FIGURA 4 **Construindo Subconjuntos de um Conjunto com $k + 1$ Elementos. Aqui, $T = S \cup \{a\}$.**

PASSO DE INDUÇÃO: Para a hipótese indutiva, assumimos que $P(k)$ seja verdadeira para o número inteiro não negativo k, ou seja, assumimos que todo conjunto com k elementos tem 2^k subconjuntos. Devemos mostrar que, considerando essa hipótese, $P(k + 1)$, que é a proposição: todo conjunto com $k + 1$ elementos tem 2^{k+1} subconjuntos, deve também ser verdadeira. Para mostrar isso, considere T como um conjunto com $k + 1$ elementos. Então, é possível escrever $T = S \cup \{a\}$, em que a é um dos elementos de T e $S = T - \{a\}$ (e, assim, $|S| = k$). Os subconjuntos de T podem ser obtidos da seguinte maneira. Para cada subconjunto X de S há exatamente dois subconjuntos de T, ou seja, X e $X \cup \{a\}$. (Conforme ilustrado na Figura 4.) Estes constituem todos os subconjuntos de T e todos são distintos. Como há 2^k subconjuntos de S, há $2 \cdot 2^k = 2^{k+1}$ subconjuntos de T. Isso finaliza o argumento indutivo.

Como completamos o passo base e o de indução, pela indução matemática temos que $P(n)$ é verdadeira para todos os números inteiros não negativos n, ou seja, demonstramos que um conjunto com n elementos tem 2^n subconjuntos sempre que n for um número inteiro não negativo. ◀

EXEMPLO 10 Use a indução matemática para demonstrar a seguinte generalização de uma das leis de De Morgan:

$$\overline{\bigcap_{j=1}^{n} A_j} = \bigcup_{j=1}^{n} \overline{A_j}$$

sempre que $A_1, A_2, \ldots, A_n$ forem subconjuntos de um conjunto universo U e $n \geq 2$.

Solução: Considere $P(n)$ como a identidade para n conjuntos.

PASSO BASE: A proposição $P(2)$ afirma que $\overline{A_1 \cap A_2} = \overline{A_1} \cup \overline{A_2}$. Esta é uma das leis de De Morgan, demonstrada na Seção 2.2.

PASSO DE INDUÇÃO: A hipótese indutiva é a proposição de que $P(k)$ seja verdadeira, em que k é um número inteiro com $k \geq 2$; ou seja, é a proposição de que

$$\overline{\bigcap_{j=1}^{k} A_j} = \bigcup_{j=1}^{k} \overline{A_j}$$

sempre que $A_1, A_2, \ldots, A_k$ forem subconjuntos do conjunto universo U. Para realizar o passo de indução, precisamos mostrar que esta hipótese implica que $P(k + 1)$ é verdadeira, ou seja, precisamos

mostrar que se esta igualdade é verdadeira para todo grupo k subconjuntos de U, então devemos também ter como verdadeira para todo grupo de $k + 1$ subconjuntos de U. Suponha que $A_1, A_2, \ldots, A_k$, A_{k+1} sejam subconjuntos de U. Quando a hipótese indutiva é assumida como verdadeira, temos que

$$\begin{aligned}
\overline{\bigcap_{j=1}^{k+1} A_j} &= \overline{\left(\bigcap_{j=1}^{k} A_j\right) \cap A_{k+1}} && \text{pela definição de intersecção} \\
&= \overline{\left(\bigcap_{j=1}^{k} A_j\right)} \cup \overline{A_{k+1}} && \text{pela lei de De Morgan (em que dois conjuntos são } \textstyle\bigcap_{j=1}^{k} A_j \text{ e } A_{k+1}) \\
&= \left(\bigcup_{j=1}^{k} \overline{A_j}\right) \cup \overline{A_{k+1}} && \text{pela hipótese indutiva} \\
&= \bigcup_{j=1}^{k+1} \overline{A_j} && \text{pela definição de união.}
\end{aligned}$$

Isso completa o passo de indução.

Como completamos o passo base e o de indução, pela indução matemática sabemos que $P(n)$ é verdadeira sempre que n for um número inteiro positivo, $n \geq 2$. Ou seja, sabemos que

$$\overline{\bigcap_{j=1}^{n} A_j} = \bigcup_{j=1}^{n} \overline{A_j}$$

sempre que $A_1, A_2, \ldots, A_n$ forem subconjuntos de um conjunto universo U e $n \geq 2$. ◀

DEMONSTRANDO RESULTADOS SOBRE ALGORITMOS Agora, fornecemos um exemplo (um pouco mais difícil que os exemplos anteriores) que ilustra uma das muitas maneiras de a indução matemática ser utilizada no estudo de algoritmos. Mostraremos que ela pode ser utilizada para demonstrar que um algoritmo "voraz" fornece uma solução ideal. (Para uma introdução sobre algoritmos "vorazes", veja a Seção 3.1.)

EXEMPLO 11 Suponha que exista um grupo de palestras com tempos pré-agendados. Gostaríamos de marcar quantas reuniões forem possíveis no salão de conferências principal. Como podemos organizar essas reuniões? Uma maneira é usar um algoritmo voraz para organizar um subconjunto com m palestras $t_1, t_2, \ldots, t_m$ na sala de conferências principal. Suponha que a reunião t_j comece no tempo b_j e termine no tempo e_j. (Duas reuniões não podem ocorrer no mesmo horário, mas uma pode começar ao mesmo tempo em que outra termine.) Assumimos que as reuniões são listadas em ordem de final de tempo não decrescente, de modo que $e_1 \leq e_2 \leq \cdots \leq e_m$. O algoritmo voraz trabalha selecionando em cada passo uma reunião que termine antes entre todas aquelas que começaram depois que todas as agendadas terminaram. Note que uma reunião que comece e termine antes é sempre selecionada em primeiro lugar pelo algoritmo. Mostraremos que este algoritmo voraz é ótimo no sentido de que sempre organiza o maior número de reuniões possíveis na sala de conferências principal. Para demonstrar suas qualidades, usamos a indução matemática na variável n, o número de reuniões agendadas pelo algoritmo. Consideremos $P(n)$ como a proposição: se o algoritmo voraz agenda n reuniões, então não é possível agendar mais que n reuniões.

PASSO BASE: Suponha que o algoritmo voraz gerencie a organização de apenas uma reunião, t_1, na sala de conferências principal. Isso significa que todas as outras reuniões não podem começar depois de e_1, o tempo final de t_1. Por outro lado, a primeira dessas reuniões é a primeira das reuniões em ordem não decrescente de tempo de término que pode ser adicionada. Assim, no tempo e_1 cada

uma das reuniões restantes precisa usar a sala de conferências porque todas começam em e_1, ou antes, e terminam depois de e_1. Temos que duas reuniões não podem ser marcadas, pois ambas precisam usar a sala no tempo e_1. Isso mostra que $P(1)$ é verdadeira e completa o passo base.

PASSO DE INDUÇÃO: A hipótese indutiva é de que $P(k)$ é verdadeira, em que k é um número inteiro positivo, ou seja, que o algoritmo voraz sempre marca o máximo de reuniões possível quando seleciona k reuniões, em que k é um número inteiro positivo, dado qualquer conjunto de reuniões, não importando o seu tamanho. Devemos mostrar que $P(k + 1)$, assumindo a hipótese de que $P(k)$ é verdadeira, ou seja, devemos mostrar que, sob essa hipótese, o algoritmo voraz sempre marca o maior número de reuniões possível quando seleciona $k + 1$ reuniões.

Agora suponha que o algoritmo selecionou $k + 1$ reuniões. Nosso primeiro passo para completar o passo de indução é mostrar que há um horário que inclui o máximo de reuniões possível que contenha t_1, uma reunião com o primeiro horário de término. Isso é fácil porque um horário que comece com a reunião t_i na lista, em que $i > 1$, pode ser trocado até que a reunião t_1 substitua t_i. Para verificar isso, note que como $e_1 \leq e_i$, todas as reuniões que foram marcadas a partir de t_i podem ainda ser marcadas.

Uma vez incluída a reunião t_1, a idéia de organizar as reuniões para que sejam marcadas quantas forem possíveis é reduzida para marcar quantas forem possíveis e que comecem em e_1 ou depois dela. Assim, se marcarmos quantas reuniões forem possíveis, o horário diferente de t_1 é um horário ideal das reuniões originais que começam quando t_1 tenha terminado. Como o algoritmo voraz organiza k reuniões quando ele cria esse horário, podemos aplicar a hipótese indutiva para concluir que ele organizou o máximo de reuniões possível. Temos que o algoritmo voraz organizou o maior número de reuniões possível, $k + 1$, quando produziu o horário com $k + 1$ reuniões, assim $P(k + 1)$ é verdadeira. Isso completa o passo de indução.

Completamos o passo base e o de indução. Assim, pela indução matemática sabemos que $P(n)$ é verdadeira para todos os números inteiros positivos n. Isso completa a demonstração das qualidades desse algoritmo, ou seja, demonstramos que quando ele organiza n reuniões, sendo n um número inteiro positivo, então não é possível marcar mais que n reuniões. ◀

USOS CRIATIVOS DE INDUÇÃO MATEMÁTICA A indução matemática pode normalmente ser utilizada de maneiras inesperadas. Vamos ilustrar dois usos particularmente inteligentes de indução matemática, o primeiro relativo a sobreviventes em uma guerra de tortas e o segundo relacionado à construção do ladrilhamento de um tabuleiro de damas.

EXEMPLO 12 **Exemplos Extras**

Guerra de Tortas Um número ímpar de pessoas está em um campo com distâncias distintas entre eles. Ao mesmo tempo, cada pessoa joga uma torta em seu vizinho mais próximo, acertando essa pessoa. Use a indução matemática para mostrar que há pelo menos um sobrevivente, ou seja, pelo menos uma pessoa que não é acertada por uma torta. (Este problema foi introduzido por [Ca79]. Note que este resultado é falso quando há um número par de pessoas; veja o Exercício 71.)

Solução: Considere $P(n)$ como a proposição de que há um sobrevivente sempre que $2n + 1$ pessoas estiverem paradas em um campo com distâncias diferentes entre elas e cada pessoa joga uma torta em seu vizinho mais próximo. Para demonstrar esse resultado, mostraremos que $P(n)$ é verdadeira para todos os números inteiros positivos n. Isso ocorre porque temos n execuções sobre todos os números inteiros positivos, $2n + 1$ execuções sobre todos os números inteiros ímpares maiores que ou iguais a 3. Note que uma pessoa não pode ser atingida em uma guerra de tortas porque não há ninguém mais para jogar a torta.

PASSO BASE: Quando $n = 1$, há $2n + 1 = 3$ pessoas na guerra de tortas. Dessas três pessoas, suponha que o par mais próximo seja A e B, e C seja a terceira pessoa. Como a distância entre os pares de pessoas é diferente, a distância entre A e C e a distância entre B e C são diferentes da distância entre A e B e maiores que ela. Temos que A joga a torta em B e B joga a torta em A, enquanto C joga a torta em A ou B, que são os mais próximos. Assim, C não é atingido por nenhuma torta. Isso mostra que pelo menos uma das três pessoas não é atingida por uma torta, completando o passo base.

PASSO DE INDUÇÃO: Para este passo, assuma que $P(k)$ seja verdadeira, ou seja, assuma que haja pelo menos um sobrevivente sempre que $2k + 1$ pessoas estejam paradas em um campo com distâncias distintas entre elas e que cada uma joga uma torta em seu vizinho mais próximo. Devemos mostrar que se a hipótese indutiva $P(k)$ for verdadeira, então $P(k + 1)$, a proposição que afirma que há pelo menos um sobrevivente sempre que $2(k + 1) + 1 = 2k + 3$ pessoas estejam paradas em um campo com distâncias diferentes entre si e cada uma joga uma torta em seu vizinho mais próximo, é também verdadeira.

Então, suponha que tenhamos $2(k + 1) + 1 = 2k + 3$ pessoas em um campo com distâncias diferentes entre os pares de pessoas. Considere A e B como o par mais próximo de pessoas nesse grupo de $2k + 3$ pessoas. Quando cada uma delas joga uma torta na pessoa mais próxima, A e B jogam as tortas entre si. Se alguém mais jogar uma torta em A ou B, então pelo menos três tortas serão jogadas em A e B e no máximo $(2k + 3) - 3 = 2k$ tortas são jogadas nas $2k + 1$ pessoas restantes. Isso garante que pelo menos uma pessoa será sobrevivente, pois se cada uma dessas $2k + 1$ pessoas foram atingidas por pelo menos uma torta, um total de pelo menos $2k + 1$ tortas foram jogadas nelas. (O argumento utilizado nesse último passo é um exemplo do princípio da casa dos pombos que será discutido mais à frente, na Seção 5.2.)

Para completar o passo de indução, suponha que ninguém mais jogue uma torta em A ou B. Além de A e B, existem $2k + 1$ pessoas. Como a distância entre os pares dessas pessoas são todas diferentes, podemos usar a hipótese indutiva para concluir que há pelo menos um sobrevivente S quando essas $2k + 1$ pessoas jogam as tortas em seus respectivos vizinhos mais próximos. Além disso, S também não é atingido pela torta jogada ou por A ou por B, porque A e B jogam suas tortas neles mesmos, então S é um sobrevivente, pois não é atingido por qualquer uma das tortas jogadas por essas $2k + 3$ pessoas. Isso completa o passo de indução e a demonstração de que $P(n)$ é verdadeira para todos os números inteiros positivos n.

Completamos o passo base e o de indução. Assim, pela indução matemática, temos que $P(n)$ é verdadeira para todos os números inteiros positivos n. Concluímos que sempre que um número ímpar de pessoas localizadas em um campo com distâncias diferentes entre si joga uma torta cada uma em seu vizinho mais próximo, existe pelo menos um sobrevivente. ◀

Links

Na Seção 1.7, discutimos o ladrilhamento de tabuleiros de damas com poliominós. O Exemplo 13 ilustra como a indução matemática pode ser utilizada para demonstrar o preenchimento de tabuleiros de damas com triominós à direita, em forma de "L".

EXEMPLO 13 Considere n como um número inteiro positivo. Mostre que todo tabuleiro de damas $2^n \times 2^n$ com um quadrado removido pode ser ladrilhado com triominós, peças que cobrem três quadrados de uma vez, como mostrado na Figura 5.

FIGURA 5
Ladrilhamento com Triominó à Direita.

Solução: Considere $P(n)$ como a proposição de que todo tabuleiro de damas $2^n \times 2^n$ com um quadrado removido pode ser ladrilhado com triominós à direita. Podemos usar a indução matemática para demonstrar que $P(n)$ é verdadeira para todos os números inteiros positivos n.

PASSO BASE: $P(1)$ é verdadeira, porque cada um dos quatro tabuleiros de damas 2×2 com um quadrado removido pode ser ladrilhado com um triominó à direita, como mostrado na Figura 6.

PASSO DE INDUÇÃO: A hipótese indutiva é a suposição de que $P(k)$ seja verdadeira para um número inteiro positivo k; ou seja, é a hipótese de que todo tabuleiro de damas $2^k \times 2^k$ com um quadrado removido possa ser ladrilhado com triominós à direita. Deve ser mostrado que,

FIGURA 6 **Ladrilhamento de Tabuleiros de Damas 2×2 com um Quadrado Removido.**

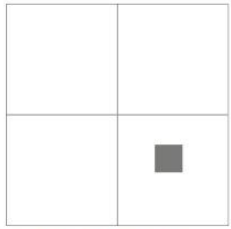


FIGURA 7 Dividindo um Tabuleiro de Damas $2^{k+1} \times 2^{k+1}$ em Quatro Tabuleiros $2^k \times 2^k$.

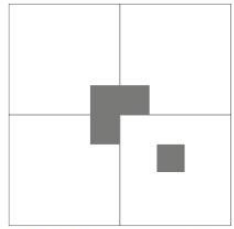


FIGURA 8 Ladrilhando um Tabuleiro de Damas $2^{k+1} \times 2^{k+1}$ com Um Quadrado Removido.

considerando-se esta hipótese indutiva, $P(k + 1)$ deve ser também verdadeira; ou seja, qualquer tabuleiro $2^{k+1} \times 2^{k+1}$ com um quadrado removido pode ser ladrilhado com triominós à direita.

Para verificar isso, considere um tabuleiro de damas $2^{k+1} \times 2^{k+1}$ com um quadrado removido. Divida esse tabuleiro em quatro tabuleiros de tamanho $2^k \times 2^k$, dividindo-o pela metade em ambas as direções. Isso está ilustrado na Figura 7. Nenhum quadrado foi removido de três dos quatro tabuleiros. O quarto tabuleiro $2^k \times 2^k$ teve um quadrado removido, assim, pela hipótese indutiva, ele pode ser ladrilhado com triominós à direita. Então, remova temporariamente o quadrado de cada um dos outros três tabuleiros $2^k \times 2^k$ que está no centro do tabuleiro original, separando uma de suas extremidades, como mostrado na Figura 8. Pela hipótese indutiva, cada um desses três tabuleiros $2^k \times 2^k$ com um quadrado removido pode ser ladrilhado, e o mesmo ocorre com os três quadrados que foram removidos temporariamente. Assim, o tabuleiro inteiro $2^{k+1} \times 2^{k+1}$ pode ser preenchido com triominós.

Completamos o passo base e o de indução. Assim, pela indução matemática, $P(n)$ é verdadeira para todos os números inteiros positivos n. Isso mostra que podemos ladrilhar todo tabuleiro $2^n \times 2^n$, em que n é um número inteiro positivo, com um quadrado removido, usando triominós à direita.

Por que a Indução Matemática é Válida

Por que a indução matemática é uma técnica de demonstração válida? A razão vem da propriedade de boa ordenação listada no Apêndice 1 como um axioma para o conjunto dos números inteiros positivos. Suponha que saibamos que $P(1)$ seja verdadeira e que a proposição $P(k) \rightarrow P(k + 1)$ seja verdadeira para todos os números inteiros positivos k. Para mostrar que $P(n)$ deve ser verdadeira para todos os números inteiros positivos n, assuma que há pelo menos um número inteiro positivo para o qual $P(n)$ seja falsa. Então, o conjunto S dos números inteiros positivos para o qual $P(n)$ seja falsa não é vazio. Assim, pela propriedade de boa ordenação, S tem pelo menos um elemento, que será indicado por m. Sabemos que m não pode ser 1, porque $P(1)$ é verdadeira. Como m é positivo e maior que 1, $m - 1$ é um número inteiro positivo. Além disso, como $m - 1$ é menor que m, ele não está em S, assim $P(m - 1)$ deve ser verdadeira. Como a proposição condicional $P(m - 1) \rightarrow P(m)$ também é verdadeira, é o caso de $P(m)$ ser verdadeira. Isso contradiz a escolha de m. Assim, $P(n)$ deve ser verdadeira para todo número inteiro positivo n.

Erros em Demonstrações que Utilizam a Indução Matemática

Como em todo método de demonstração, existem muitas oportunidades para se cometer erros. Em toda demonstração que utilize a indução matemática, devemos completar um passo base e um de indução corretos.

Note que às vezes é difícil localizar o erro em uma demonstração falsa por indução matemática, como ilustra o Exemplo 14.

EXEMPLO 14 Encontre o erro nesta "demonstração" de uma afirmação claramente falsa de que todo conjunto de retas em um plano, não paralelas duas a duas, encontram-se em um ponto comum.

"Demonstração": Considere $P(n)$ como a proposição que afirma que todo conjunto de n retas no plano, as quais não são paralelas duas a duas, encontram-se em um ponto comum. Vamos demonstrar que $P(n)$ é verdadeira para todos os números inteiros positivos $n \geq 2$.

PASSO BASE: A proposição $P(2)$ é verdadeira, pois duas retas quaisquer no plano que não são paralelas encontram-se em um ponto comum (pela definição de retas paralelas).

PASSO DE INDUÇÃO: A hipótese indutiva é a proposição que afirma que $P(k)$ é verdadeira para todo número inteiro k, ou seja, é a hipótese de que todo conjunto com k retas no plano, das quais duas delas não são paralelas, encontram-se em um ponto comum. Para completar o passo de indução, devemos mostrar que se $P(k)$ é verdadeira, então $P(k+1)$ deve ser verdadeira também, ou seja, devemos mostrar que se todo conjunto de k retas no plano, das quais duas delas não são paralelas, encontram-se em um ponto comum, então todo conjunto de $k+1$ retas no plano, das quais duas delas não são paralelas, encontram-se em um ponto comum. Assim, considere um conjunto de $k+1$ retas distintas no plano. Pela hipótese indutiva, as k primeiras dessas retas encontram-se em um ponto comum p_1. Além disso, pela hipótese indutiva, as k últimas dessas retas encontram-se em um ponto comum p_2. Mostraremos que p_1 e p_2 devem ser o mesmo ponto. Se p_1 e p_2 fossem pontos diferentes, todas as retas com ambos deveriam ser a mesma reta porque dois pontos determinam uma reta. Isso contradiz nossa hipótese de que todas essas retas são distintas. Assim, p_1 e p_2 são o mesmo ponto. Concluímos que o ponto $p_1 = p_2$ para todas as $k+1$ retas. Mostramos que $P(k+1)$ é verdadeira, assumindo que $P(k)$ seja verdadeira, ou seja, mostramos que se assumimos que todas as retas distintas k, $k \geq 2$, encontram-se em um mesmo ponto, então todas as retas distintas $k+1$ encontram-se em um ponto comum. Isso completa o passo de indução.

Completamos o passo base e o de indução e supostamente temos uma demonstração correta por indução matemática.

Solução: Examinando esta suposta demonstração por indução matemática, parece-nos que tudo está em ordem. Entretanto, há um erro particularmente sutil. Olhando com cuidado para o passo de indução, vemos que esse passo requer que $k \geq 3$. Não podemos mostrar que $P(2)$ implica $P(3)$. Quando $k = 2$, nosso objetivo é mostrar que três retas distintas quaisquer encontram-se em um ponto comum. As primeiras duas retas encontram-se em um ponto comum p_1, e as duas outras retas devem encontrar-se em um ponto comum p_2. Mas nesse caso, p_1 e p_2 não precisam ser o mesmo ponto, porque apenas a segunda reta é comum a ambos os conjuntos de retas. Aqui é onde o passo de indução falha. ◀

Exercícios

1. Há "infinitas" estações em uma rota de trem. Suponha que o trem pare na primeira estação e que, se ele pára em uma estação, pára também na próxima. Mostre que o trem pára em todas as estações.
2. Suponha que você saiba que um jogador de golfe acerta o primeiro buraco de uma seqüência com um número infinito de buracos, e que, se esse jogador acertar um buraco, então acertará o próximo também. Demonstre que esse jogador acerta todos os buracos da seqüência.

Use a indução matemática nos exercícios 3 a 17 para demonstrar as fórmulas de somatório.

3. Considere $P(n)$ como a proposição de que $1^2 + 2^2 + \cdots + n^2 = n(n+1)(2n+1)/6$ para todo número inteiro positivo n.
   a) Qual é a proposição $P(1)$?
   b) Mostre que $P(1)$ é verdadeira, completando o passo base da demonstração.
   c) Qual é a hipótese indutiva?
   d) O que você precisa para demonstrar o passo de indução?
   e) Complete o passo de indução.
   f) Explique por que esses passos mostram que esta fórmula é verdadeira sempre que n for um número inteiro positivo.
4. Considere $P(n)$ como a proposição de que $1^3 + 2^3 + \cdots + n^3 = (n(n+1)/2)^2$ para o número inteiro positivo n.
   a) Qual é a proposição $P(1)$?

b) Mostre que $P(1)$ é verdadeira, completando o passo base da demonstração.
c) Qual é a hipótese indutiva?
d) O que você precisa demonstrar no passo de indução?
e) Complete o passo de indução.
f) Explique por que esses passos mostram que esta fórmula é verdadeira sempre que n for um número inteiro positivo.

5. Demonstre que $1^2 + 3^2 + 5^2 + \cdots + (2n+1)^2 = (n+1)(2n+1)(2n+3)/3$ sempre que n for um número inteiro não negativo.

6. Demonstre que $1 \cdot 1! + 2 \cdot 2! + \cdots + n \cdot n! = (n+1)! - 1$ sempre que n for um número inteiro positivo.

7. Demonstre que $3 + 3 \cdot 5 + 3 \cdot 5^2 + \cdots + 3 \cdot 5^n = 3(5^{n+1} - 1)/4$ sempre que n for um número inteiro não negativo.

8. Demonstre que $2 - 2 \cdot 7 + 2 \cdot 7^2 - \cdots + 2(-7)^n = (1 - (-7)^{n+1})/4$ sempre que n for um número inteiro não negativo.

9. **a)** Encontre uma fórmula para a soma dos primeiros n números inteiros positivos e pares.
b) Demonstre a fórmula que você conjecturou no item (a).

10. **a)** Encontre uma fórmula para

$$\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \ldots + \frac{1}{n(n+1)}$$

examinando os valores dessa expressão para pequenos valores de n.
b) Demonstre a fórmula que você conjecturou no item (a).

11. **a)** Encontre uma fórmula para

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots + \frac{1}{2^n}$$

examinando os valores dessa expressão para pequenos valores de n.
b) Demonstre a fórmula que você conjecturou no item (a).

12. Demonstre que

$$\sum_{j=0}^{n} \left(-\frac{1}{2}\right)^j = \frac{2^{n+1} + (-1)^n}{3 \cdot 2^n}$$

sempre que n for um número inteiro não negativo.

13. Demonstre que $1^2 - 2^2 + 3^2 - \cdots + (-1)^{n-1} n^2 = (-1)^{n-1} n(n+1)/2$ sempre que n for um número inteiro positivo.

14. Demonstre que, para todo número inteiro positivo n, $\sum_{k=1}^{n} k2^k = (n-1)2^{n+1} + 2$.

15. Demonstre que, para todo número inteiro positivo n, $1 \cdot 2 + 2 \cdot 3 + \cdots + n(n+1) = n(n+1)(n+2)/3$.

16. Demonstre que, para todo número inteiro positivo n,
$1 \cdot 2 \cdot 3 + 2 \cdot 3 \cdot 4 + \cdots + n(n+1)(n+2) = n(n+1)(n+2)(n+3)/4$.

17. Demonstreque $\sum_{j=1}^{n} j^4 = n(n+1)(2n+1)\,(3n^2+3n-1)/30$ sempre que n for um número inteiro positivo.

Use a indução matemática para demonstrar as inequações nos exercícios 18 a 30.

18. Considere $P(n)$ como a proposição de que $n! < n^n$, em que n é um número inteiro maior que 1.
a) Qual é a proposição $P(2)$?
b) Mostre que $P(2)$ é verdadeira, completando o passo base da demonstração.
c) Qual é a hipótese indutiva?
d) O que você precisa para demonstrar o passo de indução?
e) Complete o passo de indução.
f) Explique por que esses passos mostram que a inequação é verdadeira sempre que n for um número inteiro maior que 1.

19. Considere $P(n)$ como a proposição de que

$$1 + \frac{1}{4} + \frac{1}{9} + \ldots + \frac{1}{n^2} < 2 - \frac{1}{n},$$

em que n é um número inteiro maior que 1.
a) Qual é a proposição $P(2)$?
b) Mostre que $P(2)$ é verdadeira, completando o passo base da demonstração.
c) Qual é a hipótese indutiva?
d) O que você precisa para demonstrar o passo de indução?
e) Complete o passo de indução.
f) Explique por que esses passos mostram que a inequação é verdadeira sempre que n for um número inteiro maior que 1.

20. Demonstre que $3^n < n!$ se n for um número inteiro maior que 6.

21. Demonstre que $2^n > n^2$ se n for um número inteiro maior que 4.

22. Para quais números inteiros não negativos n é $n^2 \leq n!$? Demonstre sua resposta.

23. Para quais números inteiros não negativos n é $2n + 3 \leq 2^n$? Demonstre sua resposta.

24. Demonstre que $1/(2n) \leq [1 \cdot 3 \cdot 5 \cdot \cdots \cdot (2n-1)]/(2 \cdot 4 \cdot \cdots \cdot 2n)$ sempre que n for um número inteiro positivo.

***25.** Demonstre que, se $h > -1$, então $1 + nh \leq (1+h)^n$ para todos os números inteiros não negativos n. Isso é chamado de **inequação de Bernoulli.**

***26.** Suponha que a e b sejam números reais com $0 < b < a$. Demonstre que se n for um número inteiro positivo, então $a^n - b^n \leq na^{n-1}(a-b)$.

***27.** Demonstre que para todo número inteiro positivo n,

$$1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \ldots + \frac{1}{\sqrt{n}} > 2(\sqrt{n+1} - 1).$$

28. Demonstre que $n^2 - 7n + 12$ é não negativo sempre que n for um número inteiro com $n \geq 3$.

Nos exercícios 29 e 30, H_n indica o n-ésimo número harmônico.

***29.** Demonstre que $H_{2^n} \leq 1 + n$ sempre que n for um número inteiro não negativo.

***30.** Demonstre que $H_1 + H_2 + \cdots + H_n = (n+1)H_n - n$.

Use a indução matemática nos exercícios 31 a 37 para demonstrar os fatos de divisibilidade.

31. Demonstre que 2 divide $n^2 + n$ sempre que n for um número inteiro positivo.

32. Demonstre que 3 divide $n^3 + 2n$ sempre que n for um número inteiro positivo.

33. Demonstre que 5 divide $n^5 - n$ sempre que n for um número inteiro não negativo.

34. Demonstre que 6 divide $n^3 - n$ sempre que n for um número inteiro não negativo.

***35.** Demonstre que $n^2 - 1$ é divisível por 8 sempre que n for um número inteiro positivo e ímpar.

***36.** Demonstre que 21 divide $4^{n+1} + 5^{2n-1}$ sempre que n for um número inteiro positivo.

***37.** Demonstre que, se n for um número inteiro positivo, então 133 divide $11^{n+1} + 12^{2n-1}$.

Use a indução matemática nos exercícios 38 a 46 para demonstrar os resultados de conjuntos.

38. Demonstre que, se $A_1, A_2, \ldots, A_n$ e $B_1, B_2, \ldots, B_n$ forem conjuntos, tal que $A_j \subseteq B_j$ para $j = 1, 2, \ldots, n$, então

$$\bigcup_{j=1}^{n} A_j \subseteq \bigcup_{j=1}^{n} B_j.$$

39. Demonstre que, se $A_1, A_2, \ldots, A_n$ e $B_1, B_2, \ldots, B_n$ forem conjuntos, tal que $A_j \subseteq B_j$ para $j = 1, 2, \ldots, n$, então

$$\bigcap_{j=1}^{n} A_j \subseteq \bigcap_{j=1}^{n} B_j.$$

40. Demonstre que, se $A_1, A_2, \ldots, A_n$ e B forem conjuntos, então

$$(A_1 \cap A_2 \cap \cdots \cap A_n) \cup B = (A_1 \cup B) \cap (A_2 \cup B) \cap \cdots \cap (A_n \cup B).$$

41. Demonstre que, se $A_1, A_2, \ldots, A_n$ e B forem conjuntos, então

$$(A_1 \cup A_2 \cup \cdots \cup A_n) \cap B = (A_1 \cap B) \cup (A_2 \cap B) \cup \cdots \cup (A_n \cap B).$$

42. Demonstre que, se $A_1, A_2, \ldots, A_n$ e B forem conjuntos, então

$$(A_1 - B) \cap (A_2 - B) \cap \cdots \cap (A_n - B) = (A_1 \cap A_2 \cap \cdots \cap A_n) - B.$$

43. Demonstre que, se $A_1, A_2, \ldots, A_n$ forem subconjuntos de um conjunto universo U, então

$$\overline{\bigcup_{k=1}^{n} A_k} = \bigcup_{k=1}^{n} \overline{A_k}.$$

44. Demonstre que, se $A_1, A_2, \ldots, A_n$ e B forem conjuntos, então

$$(A_1 - B) \cup (A_2 - B) \cup \cdots \cup (A_n - B) = (A_1 \cup A_2 \cup \cdots \cup A_n) - B.$$

45. Demonstre que um conjunto com n elementos tem $n(n-1)/2$ subconjuntos com dois elementos sempre que n for um número inteiro maior que, ou igual a 2.

***46.** Demonstre que um conjunto com n elementos tem $n(n-1)(n-2)/6$ subconjuntos com três elementos sempre que n for um número inteiro maior que ou igual a 3.

Os exercícios 47 a 49 apresentam demonstrações incorretas com o uso da indução matemática. Você precisará identificar um erro de argumento em cada exercício.

47. O que está errado nesta "demonstração" de que todos os cavalos são da mesma cor?

Considere $P(n)$ como a proposição de que todos os cavalos em um conjunto de n cavalos são da mesma cor.

Passo base: Certamente, $P(1)$ é verdadeira.

Passo de Indução: Assuma que $P(k)$ seja verdadeira, assim, todos os cavalos em qualquer conjunto de k cavalos são da mesma cor. Considere quaisquer $k+1$ cavalos; numere-os como $1, 2, 3, \ldots, k, k+1$. Agora, os primeiros k desses cavalos devem ter a mesma cor, e os últimos k cavalos devem ser também da mesma cor. Como o conjunto dos primeiros k cavalos e o conjunto dos últimos k cavalos, todos os $k+1$ cavalos devem ser da mesma cor. Isso mostra que $P(k+1)$ é verdadeira e termina a demonstração por indução.

48. O que está errado nesta "demonstração"?

"Teorema" Para todo número inteiro positivo n, $\sum_{i=1}^{n} i = (n + \frac{1}{2})^2/2$.

Passo base: A formula é verdadeira para $n = 1$.

Passo de indução: Suponha que $\sum_{i=1}^{n} i = (n + \frac{1}{2})^2/2$. Então, $\sum_{i=1}^{n+1} i = (\sum_{i=1}^{n} i) + (n+1)$. Pela hipótese indutiva, $\sum_{i=1}^{n+1} i = (n + \frac{1}{2})^2/2 + n + 1 = (n^2 + n + \frac{1}{4})/2 + n + 1 = (n^2 + 3n + \frac{9}{4})/2 = (n + \frac{3}{2})^2/2 = [(n+1) + \frac{1}{2}]^2/2$, completando o passo de indução.

49. O que está errado nesta "demonstração"?

"Teorema" Para todo número inteiro positivo n, se x e y forem números inteiros positivos com $\max(x, y) = n$, então $x = y$.

Passo base: Suponha que $n = 1$. Se $\max(x, y) = 1$ e x e y forem números inteiros positivos, temos $x = 1$ e $y = 1$.

Passo de indução: Considere k como um número inteiro positivo. Assuma que sempre que $\max(x, y) = k$ e x e y forem números inteiros positivos, então $x = y$. Agora, considere $\max(x, y) = k + 1$, em que x e y são números inteiros positivos. Então, $\max(x-1, y-1) = k$, assim, pela hipótese indutiva, $x - 1 = y - 1$. Temos que $x = y$, completando o passo de indução.

50. Use a indução matemática para mostrar que em um conjunto de $n+1$ números inteiros positivos, não excedentes a $2n$, há pelo menos um número inteiro que divide outro número inteiro do conjunto.

***51.** Um cavalo em um tabuleiro de xadrez pode se mover um espaço na horizontal (em qualquer direção) e dois espaços na vertical (em qualquer direção), ou dois espaços na horizontal (em qualquer direção) e um espaço na vertical (em qualquer direção). Suponha que tenhamos um tabuleiro de xadrez infinito, feito com todos quadrados (m, n), em que m e n são números inteiros não negativos. Use a indução matemática para mostrar que um cavalo que começa em $(0, 0)$ pode visitar todos os quadrados usando uma seqüência finita de movimentos. [*Dica:* Use a indução na variável $s = m + n$.]

52. Suponha que

$$\mathbf{A} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix},$$

em que a e b são números reais. Mostre que

$$\mathbf{A}^n = \begin{bmatrix} a^n & 0 \\ 0 & b^n \end{bmatrix}$$

para todo número inteiro positivo n.

53. (*Requer cálculo*) Use a indução matemática para demonstrar que a derivada de $f(x) = x^n$ é igual a nx^{n-1} sempre que n for um número inteiro positivo. (Para o passo de indução, use a regra do produto para derivadas.)

54. Suponha que **A** e **B** sejam matrizes quadradas com a propriedade $\mathbf{AB} = \mathbf{BA}$. Mostre que $\mathbf{AB}^n = \mathbf{B}^n\mathbf{A}$ para todo número inteiro positivo n.

55. Suponha que m seja um número inteiro positivo. Use a indução matemática para demonstrar que se a e b forem números inteiros com $a \equiv b$ (**mod** m), então $a^k \equiv b^k$ (**mod** m) sempre que k for um número inteiro não negativo.

56. Use a indução matemática para mostrar que $\neg (p_1 \vee p_2 \vee \cdots \vee p_n)$ é equivalente a $\neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n$ sempre que $p_1, p_2, \ldots, p_n$ forem proposições.

***57.** Mostre que $[(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_3) \wedge \cdots \wedge (p_{n-1} \rightarrow p_n)] \rightarrow [(p_1 \wedge p_2 \wedge \cdots \wedge p_{n-1}) \rightarrow p_n]$ é uma tautologia sempre que $p_1, p_2, \ldots, p_n$ forem proposições, em que $n \geq 2$.

***58.** Mostre que n retas separam o plano em $(n^2 + n + 2)/2$ regiões, considerando que nenhuma dessas duas retas são paralelas e que nenhuma dessas três retas passam por um ponto comum.

****59.** Considere $a_1, a_2, \ldots, a_n$ como números reais positivos. A **média aritmética** desses números é definida por $A = (a_1 + a_2 + \cdots + a_n)/n$, e a **média geométrica** desses números é definida por $G = (a_1 a_2 \cdots a_n)^{1/n}$. Use a indução matemática para demonstrar que $A \geq G$.

60. Use a indução matemática para demonstrar o Lema 2 da Seção 3.6, que estabelece que se p é um número primo e $P \mid a_1 a_2 \cdots a_n$, em que a_i é um número inteiro para $i = 1, 2, 3, \cdots, n$, então $p \mid a_i$ para algum número inteiro i.

61. Mostre que se n for um número inteiro positivo, então

$$\sum_{\{a_1, \ldots, a_k\} \subseteq \{1, 2, \ldots, n\}} \frac{1}{a_1 a_2 \ldots a_k} = n.$$

(Aqui, a soma é sobre todos os subconjuntos não vazios do conjunto dos n menores números inteiros positivos.)

***62.** Use a propriedade da boa ordenação para mostrar que a seguinte forma da indução matemática é um método de demonstração válido: $P(n)$ é verdadeira para todos os números inteiros positivos n.

Passo Base: $P(1)$ e $P(2)$ são verdadeiras.
Passo de indução: Para cada número inteiro positivo k, se $P(k)$ e $P(k+1)$ forem verdadeiras, então $P(k+2)$ é verdadeira.

63. Mostre que, se $A_1, A_2, \ldots, A_n$ forem conjuntos em que $n \geq 2$, e para todos os pares de números inteiros i e j com $1 \leq i < j \leq n$, ou A_i é um subconjunto de A_j ou A_j é um subconjunto de A_i, então existe um número inteiro i, $1 \leq i \leq n$, tal que A_i é um subconjunto de A_j para todos os números inteiros j com $1 \leq j \leq n$.

***64.** Um convidado em uma festa é uma **celebridade** se essa pessoa for conhecida por todos os outros convidados, mas não conhecer nenhum deles. Existe no máximo uma celebridade em uma festa, pois, se tiverem duas, elas deveriam conhecer uma a outra. Determinada festa não pode ter celebridades. Sua tarefa é encontrar a celebridade, se ela existir, levantando apenas um tipo de questão — perguntando aos convidados se eles conhecem um segundo convidado. Todos devem responder à questão sem mentir, ou seja, se Alice e Bob forem duas pessoas na festa, você pode perguntar a Alice se ela conhece Bob, e ela deve dizer a verdade. Use a indução matemática para mostrar que se existirem n pessoas na festa, então você pode encontrar a celebridade, se houver uma, com $3(n-1)$ perguntas. [*Dica:* Primeiro faça uma questão para eliminar uma pessoa como celebridade. Então, use a hipótese indutiva para identificar uma celebridade em potencial. Por fim, faça mais duas questões para determinar se aquela pessoa é realmente uma celebridade.]

Suponha que haja n pessoas em um grupo, cada uma ciente de um escândalo que ninguém mais no grupo sabe. Essas pessoas comunicam-se por telefone; quando duas pessoas do grupo conversam, elas compartilham informações sobre todos os escândalos que cada uma sabe. Por exemplo, na primeira chamada, duas pessoas compartilham informações, assim, no final desta chamada, cada uma dessas pessoas sabe sobre dois escândalos. O **problema da fofoca** pergunta por $G(n)$, o número mínimo de ligações telefônicas que são necessárias para que todas as n pessoas saibam sobre todos os escândalos. Os exercícios 65 a 67 trabalham com o problema da fofoca.

65. Encontre $G(1)$, $G(2)$, $G(3)$ e $G(4)$.

66. Use a indução matemática para demonstrar que $G(n) \leq 2n - 4$ para $n \geq 4$. [*Dica:* No passo de indução, coloque uma nova pessoa que liga para determinada pessoa no começo e no final.]

****67.** Demonstre que $G(n) = 2n - 4$ para $n \geq 4$.

***68.** Mostre que é possível organizar os números $1, 2, \ldots, n$ em uma linha para que a média de dois desses números nunca apareça entre eles. [*Dica:* Mostre que é suficiente demonstrar esse fato quando n é uma potência de 2. Então, use a indução matemática para demonstrar o resultado quando n for uma potência de 2.]

***69.** Mostre que se $I_1, I_2, \ldots, I_n$ for um grupo de intervalos abertos de números reais, $n \geq 2$, e cada par de intervalos tiver uma intersecção não vazia, ou seja, $I_i \cap I_j \neq \emptyset$ sempre que $1 \leq i \leq n$ e $1 \leq j \leq n$, então a intersecção de todos esses conjuntos é não vazia, ou seja, $I_1 \cap I_2 \cap \cdots \cap I_n \neq \emptyset$. (Lembre-se de que um **intervalo aberto** é o conjunto dos números reais x com $a < x < b$, em que a e b são números reais com $a < b$.)

Às vezes não podemos usar a indução matemática para demonstrar um resultado que acreditamos ser verdadeiro, mas podemos usá-la para demonstrar um resultado mais forte. Como a hipótese indutiva de um resultado mais forte fornece mais do que trabalhar com ele, esse processo é chamado de **carga indutiva.** Podemos usar a carga indutiva no Exercício 70.

70. Suponha que queiramos demonstrar que

$$\frac{1}{2} \cdot \frac{3}{4} \cdots \frac{2n-1}{2n} < \frac{1}{\sqrt{3n}}$$

para todos os números inteiros positivos n.

a) Mostre que, se tentarmos demonstrar esta inequação usando a indução matemática, o passo base será válido, mas o de indução não.

b) Mostre que a indução matemática pode ser utilizada para demonstrar a inequação forte

$$\frac{1}{2}\cdot\frac{3}{4}\cdots\frac{2n-1}{2n} < \frac{1}{\sqrt{3n+1}}$$

para todos os números inteiros maiores que 1, que, junto com a verificação do caso em que $n = 1$, estabelece a inequação mais fraca que originariamente tentamos demonstrar usando a indução matemática.

71. Considere n como um número inteiro positivo par. Mostre que, quando n pessoas ficam paradas em um campo com distâncias diferentes entre elas e cada uma joga uma torta no seu vizinho mais próximo, é possível que todos sejam atingidos por uma torta.

72. Ladrilhe com triominós à direita um tabuleiro de damas 4×4 com um quadrado removido na parte superior esquerda.

73. Ladrilhe com triominós à direita um tabuleiro de damas 8×8 com um quadrado removido na parte superior esquerda.

74. Demonstre ou negue que todos os tabuleiros de damas nos tamanhos a seguir podem ser completamente preenchidos com triominós à direita sempre que n for um número inteiro positivo.

a) 3×2^n. **b)** 6×2^n.

c) $3^n \times 3^n$. **d)** $6^n \times 6^n$.

***75.** Mostre que um tabuleiro de damas $2^n \times 2^n \times 2^n$ tridimensional, em que falta um cubo $1 \times 1 \times 1$, pode ser completamente preenchido com cubos $2 \times 2 \times 2$, considerando-se que um cubo $1 \times 1 \times 1$ tenha sido removido.

***76.** Mostre que um tabuleiro de damas $n \times n$ com um quadrado removido pode ser completamente preenchido com triominós, se $n > 5$, n for ímpar e $3 \nmid n$.

77. Mostre que um tabuleiro de damas 5×5 com um quadrado do canto removido pode ser ladrilhado com triominós.

***78.** Encontre um tabuleiro de damas 5×5 com um quadrado removido que não pode ser ladrilhado com triominós. Demonstre que esse preenchimento não existe para esse tipo de tabuleiro.

☞**79.** Use o princípio da indução matemática para mostrar que $P(n)$ é verdadeira, para $n = b, b + 1, b + 2, \ldots$, em que b é um número inteiro, se $P(b)$ for verdadeira e a proposição condicional $P(k) \rightarrow P(k + 1)$ for verdadeira para todos os números inteiros positivos k com $k \geq b$.

4.2 Indução Completa e Boa Ordenação

Introdução

Na Seção 4.1, introduzimos a indução matemática e mostramos como usá-la para demonstrar vários teoremas. Nesta seção, vamos introduzir outra forma de indução matemática, chamada de **indução completa**, que geralmente pode ser utilizada quando não podemos demonstrar facilmente um resultado com a indução matemática. O passo base de uma demonstração por indução completa é o mesmo de uma demonstração por indução matemática, ou seja, em uma demonstração por indução completa em que $P(n)$ é verdadeira para todos os números inteiros positivos n, o passo base mostra que $P(1)$ é verdadeira. Entretanto, os passos de indução nos dois métodos são diferentes. Em uma demonstração por indução matemática, o passo de indução mostra que se a hipótese indutiva $P(k)$ for verdadeira, então $P(k + 1)$ será também verdadeira. Em uma demonstração por indução completa, o passo de indução mostra que se $P(j)$ for verdadeira para todos os números inteiros positivos j não excedentes a k, então $P(k + 1)$ é verdadeira, ou seja, para a hipótese indutiva, assumimos que $P(j)$ seja verdadeira para $j = 1, 2, \ldots, k$.

A validade da indução matemática e da indução completa seguem da propriedade da boa ordenação, descrita no Apêndice 1. De fato, a indução matemática, a indução completa e a boa ordenação são princípios equivalentes, ou seja, a validade de cada um deles pode ser demonstrada a partir de cada um dos outros dois. Isso significa que uma demonstração que usa um desses princípios pode ser reescrita como uma demonstração que usa os outros dois princípios. Por isso, às vezes é mais fácil demonstrar um resultado usando a indução completa em vez da indução matemática; às vezes é mais fácil usar a boa ordenação em vez de usar essas duas outras formas de indução. Nesta seção daremos alguns exemplos da utilização da propriedade da boa ordenação para demonstrar teoremas.

Indução Completa

Antes de ilustrarmos como usar a indução completa, vamos estabelecer este princípio novamente.

INDUÇÃO COMPLETA Para demonstrar que $P(n)$ é verdadeira para todos os números inteiros positivos n, em que $P(n)$ é uma função proposicional, completamos dois passos:

PASSO BASE: Verificamos que a proposição $P(1)$ é verdadeira.

PASSO DE INDUÇÃO: Mostramos que a proposição condicional $[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$ é verdadeira para todos os números inteiros positivos k.

Note que quando usamos a indução completa para demonstrar que $P(n)$ é verdadeira para todos os números inteiros positivos n, nossa hipótese indutiva é de que $P(j)$ seja verdadeira para $j = 1, 2, \ldots, k$, ou seja, a hipótese indutiva inclui todas as k proposições $P(1), P(2), \ldots, P(k)$. Como podemos usar todas as k proposições $P(1), P(2), \ldots, P(k)$ para demonstrar $P(k+1)$, em vez de apenas usar a proposição $P(k)$, como na demonstração por indução matemática, a indução completa é uma técnica de demonstração mais flexível. Você pode ficar surpreso pelo fato de a indução matemática e a completa serem equivalentes, ou seja, cada uma pode ser mostrada como uma técnica de demonstração válida, assumindo que a outra é válida. Em particular, qualquer demonstração que use a indução matemática também pode ser considerada como uma demonstração por indução completa, porque a hipótese indutiva de uma demonstração por indução matemática é parte da hipótese indutiva em uma demonstração por indução completa. Ou seja, se podemos completar a etapa indutiva de uma demonstração usando a indução matemática ao mostrar que $P(k+1)$ segue de $P(k)$ para todo número inteiro positivo k, então temos também que $P(k+1)$ segue de todas as proposições $P(1), P(2), \ldots, P(k)$, porque assumimos que não apenas $P(k)$ é verdadeira, mas também que as $k-1$ proposições $P(1), P(2), \ldots, P(k-1)$ são verdadeiras. Entretanto, é muito mais difícil converter uma demonstração por indução completa em uma demonstração que use o princípio da indução matemática. (Veja o Exercício 42.)

A indução completa às vezes é chamada de **segundo princípio da indução matemática** ou **indução forte.** Quando a terminologia "indução completa" é utilizada, o princípio da indução matemática é chamado de **indução incompleta**, um termo técnico escolhido de modo infeliz porque não há nada de incompleto no princípio da indução matemática; além disso, ela é uma técnica de demonstração válida.

INDUÇÃO COMPLETA E A ESCADA INFINITA Para melhor entender a indução completa, considere a escada infinita da Seção 4.1. A indução completa nos diz que podemos alcançar todos os degraus se

1. pudermos atingir o primeiro degrau e
2. para todo número inteiro k, se pudermos alcançar todos os primeiros k degraus, então podemos alcançar o $(k+1)$-ésimo degrau.

Ou seja, se $P(n)$ for a proposição que afirma que podemos alcançar o n-ésimo degrau da escada, pela indução completa, sabemos que $P(n)$ é verdadeira para todos os números inteiros positivos n, pois (1) nos diz que $P(1)$ é verdadeira, completando o passo base, e (2) nos diz que $P(1) \wedge P(2) \wedge \cdots \wedge P(k)$ implica $P(k+1)$, completando o passo de indução.

O Exemplo 1 ilustra como a indução completa pode nos ajudar a demonstrar um resultado que não pode ser facilmente demonstrado com o princípio da indução matemática.

EXEMPLO 1 Suponha que nós podemos alcançar o primeiro e o segundo degraus de uma escada infinita e sabemos que se pudermos alcançar um degrau, então podemos alcançar dois degraus acima. Podemos demonstrar que é possível alcançar todos os degraus usando o princípio da indução matemática? Podemos demonstrar que é possível alcançar todos os degraus usando o princípio da indução completa?

Solução: Primeiro, tentamos demonstrar esse resultado pelo princípio da indução matemática.

PASSO BASE: O passo base dessa demonstração é válido; aqui, simplesmente é verificado que nós podemos alcançar o primeiro degrau.

TENTATIVA DE PASSO DE INDUÇÃO: A hipótese indutiva é a proposição que afirma que podemos alcançar o k-ésimo degrau da escada. Para completar o passo de indução, precisamos mostrar que se assumirmos a hipótese indutiva para o número inteiro positivo k, ou seja, se assumirmos que podemos alcançar o k-ésimo degrau, então podemos mostrar que é possível alcançar o $(k + 1)$-ésimo degrau da escada. Entretanto, não existe uma forma óbvia de completar esse passo de indução, porque não sabemos a partir da informação dada se é possível alcançar o $(k + 1)$-ésimo degrau a partir do k-ésimo degrau. Além disso, sabemos apenas que se podemos alcançar um degrau, podemos alcançar também dois degraus acima.

Agora, considere uma demonstração que usa a indução completa.

PASSO BASE: O passo base é igual ao primeiro; aqui, simplesmente, é verificado que podemos alcançar o primeiro degrau.

PASSO DE INDUÇÃO: A hipótese indutiva afirma que podemos alcançar cada um dos primeiros k degraus. Para completá-la, precisamos mostrar que se assumirmos que a hipótese indutiva é verdadeira, ou seja, se pudermos alcançar cada um dos primeiros k degraus, então podemos alcançar o $(k + 1)$-ésimo degrau. Nós já sabemos que é possível alcançar o segundo degrau. Podemos completar o passo de indução notando que enquanto $k > 2$, será possível alcançar o $(k + 1)$-ésimo degrau a partir do $(k - 1)$-ésimo degrau, porque sabemos que podemos subir dois degraus a partir daquele em que nos encontramos. Isso completa o passo de indução e termina a demonstração por indução completa.

Demonstramos que se podemos alcançar os dois primeiros degraus de uma escada infinita e para todo número inteiro k, se pudermos alcançar todos os primeiros k degraus, então podemos alcançar o $(k + 1)$-ésimo degrau e, assim, alcançar todos os degraus da escada. ◀

Exemplos de Demonstrações com o Uso da Indução Completa

Agora que temos a indução matemática e a indução completa, como decidir qual método aplicar em determinada situação? Embora não haja uma resposta direta, podemos fornecer alguns pontos úteis. Na prática, você deve usar a indução matemática quando conseguir uma demonstração direta de $P(k) \rightarrow P(k + 1)$ para todos os números inteiros k. Este é o caso de todas as demonstrações nos exemplos da Seção 4.1. Em geral, você deve restringir o uso do princípio da indução matemática para esses casos. A menos que veja claramente que o passo de indução de uma demonstração por indução matemática é válido, você deve tentar uma demonstração por indução completa. Ou seja, use a indução completa, e não a matemática, quando perceber como demonstrar que $P(k + 1)$ é verdadeira a partir da hipótese que afirma que $P(j)$ é verdadeira para todos os números inteiros positivos j, não excedentes a k, mas não perceber como demonstrar que $P(k + 1)$ é conseqüência apenas de $P(k)$. Lembre-se disso ao examinar as demonstrações desta seção. Para cada uma dessas demonstrações, pense por que a indução completa funciona melhor que a indução matemática.

Vamos ilustrar como a indução completa é emprega nos exemplos 2 a 5, nos quais demonstraremos vários resultados diferentes. Fique atento particularmente ao passo de indução em cada um desses exemplos, onde mostramos um resultado $P(k + 1)$, considerando a hipótese de que $P(j)$ é verdadeira para todos os números inteiros positivos j não excedentes a k, em que $P(n)$ é uma função proposicional.

Começamos com um dos usos mais importantes da indução completa, a parte do Teorema Fundamental da Aritmética que nos diz que todo número inteiro positivo pode ser escrito como o produto de números primos.

EXEMPLO 2 Mostre que se n for um número inteiro maior que 1, então n pode ser escrito como o produto de números primos.

Exemplos Extras

Solução: Considere $P(n)$ como a proposição: n pode ser escrito como o produto de números primos.

PASSO BASE: $P(2)$ é verdadeira, porque 2 pode ser escrito como o produto de um número primo, ele mesmo. [Note que $P(2)$ é o primeiro caso que precisamos estabelecer.]

PASSO DE INDUÇÃO: A hipótese indutiva estabelece que $P(j)$ é verdadeira para todos os números inteiros positivos j com $j \leq k$, ou seja, a hipótese de que j possa ser escrito como o produto de números primos sempre que j for um número inteiro positivo, ao menos 2 e não excedendo a k. Para completar o passo de indução, devemos mostrar que $P(k + 1)$ é verdadeira, considerando esta hipótese, ou seja, que $k + 1$ é o produto de números primos.

Há dois casos a serem considerados, quando $k + 1$ é primo e quando $k + 1$ é composto. Se $k + 1$ for primo, imediatamente vemos que $P(k + 1)$ é verdadeira. Por outro lado, $k + 1$ é composto, e pode ser escrito como o produto de dois números inteiros positivos a e b com $2 \leq a \leq b < k + 1$. Pela hipótese indutiva, a e b podem ser escritos como o produto de números primos. Assim, se $k + 1$ for composto, ele pode ser escrito como o produto de números primos, quais sejam, aqueles primos obtidos na fatoração de a e aqueles na fatoração de b. ◀

Lembre-se: Como é possível pensar em 1 como o produto *vazio* de primos, poderíamos ter começado a demonstração do Exemplo 2 com $P(1)$ como o passo base. Escolhemos não fazer isso porque muitas pessoas acham esse procedimento confuso.

O Exemplo 2 completa a demonstração do Teorema Fundamental da Aritmética, que afirma que todo número inteiro não negativo pode ser escrito unicamente como o produto de números inteiros em ordem não decrescente. Mostramos na Seção 3.5 que um número inteiro tem no máximo uma fatoração em números primos. O Exemplo mostra que existe pelo menos uma fatoração para todos os inteiros.

Agora, mostraremos que a indução completa pode ser utilizada para demonstrar que um jogador tem uma estratégia para ganhar em um jogo.

EXEMPLO 3 Considere um jogo em que dois jogadores alternam-se para remover qualquer número positivo de cartas que eles pegam a partir de dois montes de cartas de baralho. O jogador que tirar a última carta, ganha o jogo. Mostre que se duas pilhas contêm o mesmo número de cartas inicialmente, o segundo jogador sempre ganha o jogo, ou tem uma estratégia para isso.

Solução: Considere n como o número de cartas em cada pilha. Vamos usar a indução completa para demonstrar $P(n)$, a proposição que afirma que o segundo jogador pode vencer quando houver inicialmente n cartas em cada monte.

PASSO BASE: Quando $n = 1$, o primeiro jogador tem apenas uma escolha, remover uma carta de um monte, deixando um monte com apenas uma carta, que o segundo jogador pode retirar para vencer o jogo.

PASSO DE INDUÇÃO: A hipótese indutiva é a proposição: $P(j)$ é verdadeira para todo j com $1 \leq j \leq k$, ou seja, a hipótese de que o segundo jogador pode vencer sempre que houver j cartas, em que $1 \leq j \leq k$ em cada um dos dois montes no início do jogo. Precisamos mostrar que $P(k + 1)$ é verdadeira, ou seja, que o segundo jogador pode vencer quando há inicialmente $k + 1$ cartas em cada monte, considerando a hipótese que $P(j)$ é verdadeira para $j = 1, 2, \ldots, k$. Então, suponha que haja $k + 1$ cartas em cada um dos dois montes no início do jogo e que o primeiro jogador retira r cartas ($1 \leq r \leq k$) a partir de um dos montes, deixando $k + 1 - r$ cartas nesse monte. Ao remover o mesmo número de cartas do outro monte, o segundo jogador cria a situação em que há dois montes, cada um com $k + 1 - r$ cartas. Como $1 \leq k + 1 - r \leq k$, o segundo jogador sempre vence pela hipótese indutiva. Completamos a demonstração notando que se o primeiro jogador remover todas as $k + 1$ cartas de um dos montes, o segundo jogador pode vencer removendo todas as cartas restantes. ◀

Usar o princípio da indução matemática, em vez da indução completa, para demonstrar os resultados dos exemplos 2 e 3 é muito difícil. Entretanto, como mostra o Exemplo 4, alguns resultados podem ser demonstrados usando-se o princípio da indução matemática ou o da indução completa.

Antes de apresentarmos o Exemplo 4, note que podemos modificar a indução completa para lidar com várias situações. Em particular, podemos adaptá-la para lidar com casos em que o passo de indução é válido apenas para números inteiros maiores que determinado inteiro. Considere b como um número inteiro e j como outro número inteiro positivo. A forma da indução completa que precisamos nos diz que $P(n)$ é verdadeira para todos os números inteiros n com $n \geq b$ se pudermos completar as duas etapas abaixo:

PASSO BASE: Verificamos que as proposições $P(b), P(b+1), \cdots, P(b+j)$ são verdadeiras.

PASSO DE INDUÇÃO: Mostramos que $[P(b) \wedge P(b+1) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$ é verdadeira para todo número inteiro positivo $k \geq b+j$.

Vamos usar essa forma alternativa de demonstração por indução completa no Exemplo 4. No Exercício 28, mostraremos que essa forma alternativa é equivalente à indução completa.

EXEMPLO 4 Demonstre que qualquer valor de postagem maior que ou igual a 12 centavos pode ser obtido usando-se apenas selos de 4 e 5 centavos.

Solução: Vamos demonstrar este resultado usando o princípio da indução matemática e depois, o da indução completa. Considere $P(n)$ como a proposição que afirma que a postagem de n centavos pode ser feita usando-se selos de 4 e 5 centavos.

Começamos usando o princípio de indução matemática.

PASSO BASE: A postagem de 12 centavos pode ser feita usando-se três selos de 4 centavos.

PASSO DE INDUÇÃO: A hipótese indutiva é a proposição que afirma que $P(k)$ é verdadeira. Ou seja, considerando-se esta hipótese, a postagem de k centavos pode ser feita usando-se selos de 4 e 5 centavos. Para completar o passo de indução, precisamos mostrar que quando assumimos que $P(k)$ é verdadeira, então $P(k+1)$ é também verdadeira em que $k \geq 12$, ou seja, precisamos mostrar que se pudermos fazer a postagem de k centavos, então faremos a postagem de $k+1$ centavos. Para isso, suponha que pelo menos um selo de 4 centavos foi usado para a postagem de k centavos. Então, podemos substituir este selo por um de 5 centavos para a postagem de $k+1$ centavos. Mas, se nenhum selo de 4 centavos tiver sido utilizado, podemos fazer a postagem de k centavos usando apenas selos de 5 centavos. Além disso, como $k \geq 12$, precisamos de pelo menos três selos de 5 centavos para formar a postagem de k centavos. Assim, podemos substituir os três selos de 5 centavos por quatro selos de 4 centavos para fazer a postagem de $k+1$ centavos. Isso completa a etapa indutiva.

Como nós completamos o passo base e o de indução, sabemos que $P(n)$ é verdadeira para todo $n \geq 12$. Ou seja, podemos fazer postagens de n centavos, em que $n \geq 12$, usando apenas selos de 4 e 5 centavos.

Agora, usaremos a indução completa para demonstrar o mesmo resultado. Nesta demonstração, no passo base, mostraremos que $P(12)$, $P(13)$, $P(14)$ e $P(15)$ são verdadeiras, ou seja, que a postagem de 12, 13, 14 ou 15 centavos pode ser feita usando apenas selos de 4 e 5 centavos. No passo de indução mostraremos como conseguir postagens de $k+1$ centavos para $k \geq 15$ a partir de postagens de $k-3$ centavos.

PASSO BASE: Podemos formar postagens de 12, 13, 14 e 15 centavos usando três selos de 4 centavos, dois selos de 4 centavos e um selo de 5 centavos, um selo de 4 centavos e dois de 5 centavos, e três selos de 5 centavos, respectivamente. Isso mostra que $P(12)$, $P(13)$, $P(14)$ e $P(15)$ são verdadeiras e completa o passo base.

PASSO DE INDUÇÃO: A hipótese indutiva é a proposição: $P(j)$ é verdadeira para $12 \leq j \leq k$, em que k é um número inteiro com $k \geq 15$. Ou seja, assumimos que podemos fazer postagens de j centavos, em que $12 \leq j \leq k$. Para completar o passo de indução, precisamos mostrar que, considerando-se esta hipótese, $P(k+1)$ é verdadeira, ou seja, podemos fazer a postagem de $k+1$ centavos. Usando a hipótese indutiva, podemos assumir que $P(k-3)$ é verdadeira, porque $k-3 \geq 12$, ou seja, podemos fazer postagens de $k-3$ centavos usando apenas selos de 4 e 5 centavos. Para fazer a postagem de $k+1$ centavos, precisamos apenas adicionar outro selo de 4 centavos aos selos que usamos para fazer a postagem de $k-3$ centavos. Ou seja, mostramos que se a hipótese indutiva for verdadeira, então $P(k+1)$ será também verdadeira. Isso completa o passo de indução.

Como completamos o passo base e o de indução da demonstração por indução completa, sabemos, por esta indução, que $P(n)$ é verdadeira para todos os números inteiros n com $n \geq 12$. Ou seja, sabemos que toda postagem de n centavos, em que n é ao menos 12, pode ser feita com selos de 4 e 5 centavos.

(Existem outras maneiras de abordar este problema além desta aqui descrita. Você pode encontrar uma solução que não usa a indução matemática?) ◀

Usando a Indução Completa na Geometria Computacional

Nosso próximo exemplo de indução completa virá da **geometria computacional**, a parte da matemática discreta que estuda os problemas de computação que envolvem objetos geométricos. A geometria computacional é muito utilizada em gráficos, jogos de computador, robótica, cálculo científico e em muitas outras áreas. Antes de apresentarmos este assunto, vamos introduzir alguns termos, possivelmente familiares, de estudos anteriores sobre geometria.

Um **polígono** é uma figura geométrica fechada que consiste em uma seqüência de segmentos de retas $s_1, s_2, \ldots, s_n$, chamados de **lados.** Cada par de lados consecutivos, s_i e s_{i+1}, $i = 1, 2, \ldots, n-1$, assim como o último lado s_n com o primeiro lado s_1, de um polígono encontram-se em um ponto comum, chamado de **vértice.** Um polígono é chamado de **simples** se não houver intersecção entre dois lados não consecutivos. Todo polígono simples divide o plano em duas regiões: seu **interior**, que são os pontos localizados dentro da curva, e o seu **exterior**, que são os pontos de fora da curva. Este fato é surpreendentemente complicado de ser demonstrado. É um caso especial do famoso Teorema da Curva de Jordan, que nos diz que toda curva simples divide o plano em duas regiões; veja [Or00] para mais exemplos.

Um polígono é chamado de **convexo** se todo segmento de reta que ligar dois pontos em seu interior estiver totalmente dentro do polígono. (Um polígono que não é convexo é chamado de **não convexo**.) A Figura 1 mostra alguns polígonos; (a) e (b) são convexos, mas (c) e (d) não. A **diagonal** de um polígono simples é um segmento de reta que liga dois vértices não consecutivos, e uma diagonal é chamada de **diagonal interna** se ela estiver totalmente dentro do polígono, exceto pelos seus pontos extremos. Por exemplo, no polígono (d), o segmento de reta que liga a e f é uma diagonal interna, mas o segmento de reta que liga a e d é uma diagonal que não é interna.

Uma das operações mais básicas da geometria computacional envolve a divisão de um polígono simples em triângulos, adicionando-se diagonais que não se cruzam. Este processo é chamado de **triangulação.**

Note que um polígono simples pode ter muitas triangulações diferentes, como mostra a Figura 2. Talvez o fato mais básico na geometria computacional seja aquele que afirma que é possível triangular todo polígono simples, como afirma o Teorema 1. Além disso, esse teorema nos diz que toda triangulação de um polígono simples com n lados inclui $n - 2$ triângulos.

TEOREMA 1 Um polígono simples com n lados, em que n é um número inteiro com $n \geq 3$, pode ser triangulado em $n - 2$ triângulos.

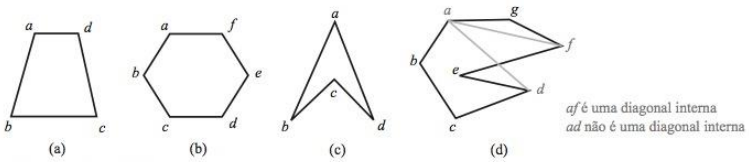
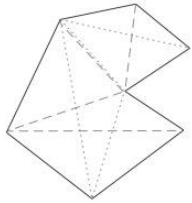


FIGURA 1 **Polígonos Convexos e Não Convexos.**



Duas triangulações diferentes de um polígono simples com sete lados em cinco triângulos, mostrados com linhas pontilhadas e tracejadas, respectivamente

FIGURA 2 Triangulações de um Polígono.

Parece óbvio que devemos ser capazes de fazer a triangulação de um polígono simples a partir de adições sucessivas de diagonais internas. Conseqüentemente, uma demonstração por indução completa parece promissora. Entretanto, essa demonstração requer este lema crucial.

LEMA 1 Todo polígono simples tem uma diagonal interna.

Embora o Lema 1 pareça ser simples, é surpreendentemente difícil de ser demonstrado. Nos últimos 30 anos, várias demonstrações incorretas que pensava-se estarem corretas, eram vistas comumente em artigos e livros. Demonstraremos o Lema 1 depois de demonstrar o Teorema 1. Não é incomum demonstrar um teorema que está pendente para depois demonstrar um lema importante.

Demonstração (do Teorema 1): Vamos demonstrar este resultado usando a indução completa. Considere $T(n)$ como a proposição que afirma que todo polígono simples com n lados pode ser triangulado em $n - 2$ triângulos.

PASSO BASE: $T(3)$ é verdadeira porque um polígono simples com três lados é um triângulo. Não precisamos adicionar nenhuma diagonal para triangular um triângulo; ele já esta triangulado em um triângulo, ele mesmo.

PASSO DE INDUÇÃO: Para a hipótese indutiva, assumimos que $T(j)$ seja verdadeira para todos os números inteiros j com $3 \leq j \leq k$. Ou seja, assumimos que podemos triangular um polígono simples com j lados em $j - 2$ triângulos sempre que $3 \leq j \leq k$. Para completar o ponto de indução, devemos mostrar que quando assumimos a hipótese indutiva, $P(k + 1)$ é verdadeira, ou seja, que todo polígono simples com $k + 1$ lados pode ser triangulado em $(k + 1) - 2 = k - 1$ triângulos.

Então, suponha que tenhamos um polígono simples P com $k + 1$ lados. Pelo Lema 1, P tem uma diagonal interna ab. Agora, ab divide P em dois polígonos simples Q, com s lados, e R, com t lados. Os lados de Q e R são os lados de P, junto com o lado ab, que é um lado tanto de Q quanto de R. Note que $3 \leq s \leq k$ e $3 \leq t \leq k$, pois Q e R têm pelo menos um lado a menos que P (uma vez que cada um desses lados é formado por P apagando pelo menos dois lados e substituindo-os pela diagonal ab). Além disso, o número de lados de P é dois a menos que a soma dos números de lados de Q e o número de lados de R, pois cada lado de P é um lado de Q ou de R, mas não de ambos, e a diagonal ab é um lado tanto de Q quanto de R, mas não de P. Ou seja, $k + 1 = s + t - 2$.

Pela hipótese indutiva, como $3 \leq s \leq k$ e $3 \leq t \leq k$, podemos triangular Q e R em $s - 2$ e $t - 2$ triângulos, respectivamente. Agora, note que essas triangulações juntas produzem uma triangulação de P. (Cada diagonal adicionada para triangular um desses polígonos menores é também uma diagonal de P.) Conseqüentemente, podemos triangular P em um total de $(s - 2) + (t - 2) = s + t - 4 = (k + 1) - 2$ triângulos. Isso completa a demonstração por indução completa, ou seja, mostramos que todo polígono simples com n lados, em que $n \geq 3$, pode ser triangulado em $n - 2$ triângulos. ◁

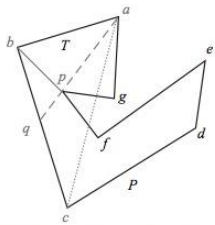


FIGURA 3 Construindo uma Diagonal Interna de um Polígono Simples.

Retomaremos agora nossa demonstração do Lema 1. Apresentaremos uma demonstração publicada por Chung-Wu Ho [Ho75]. Note que embora essa demonstração possa ser omitida sem perda de continuidade, mostraremos como pode ser difícil demonstrar um resultado que parece ser um pouco óbvio.

Demonstração: Suponha que P seja um polígono simples em um plano. Além disso, suponha que b seja o ponto de P ou do interior de P com a menor coordenada y entre os vértices com a menor coordenada x. Então, b deve ser um vértice de P, pois se for um ponto interno, deverá ter um vértice de P com a menor coordenada x. Dois outros vértices consecutivos a b: a e c, formam um ângulo b. Daqui temos que o ângulo no interior de P formado por ab e bc deve ser menor que 180 graus (caso contrário, deveriam haver pontos de P com coordenadas x menores que b).

Agora, considere T como o triângulo Δabc. Se não existirem vértices de P sobre ou no interior de T, podemos ligar a e c para obter uma diagonal interna. Por outro lado, se houver vértices de P no interior de T, encontraremos um vértice p de P sobre ou no interior de T, tal que bp é uma diagonal interna. (Esta é a parte genial. Ho notou que em muitas demonstrações publicadas sobre esse lema, um vértice p era encontrado, tal que bp não era necessariamente uma diagonal interna de P. Veja o Exercício 21.) A chave é selecionar um vértice p, tal que o ângulo $\angle bap$ seja o menor. Para isso, note que o raio que começa em a e passa por p atinge o segmento de reta bc em um ponto, digamos q. Temos, então, que o triângulo Δbaq não pode conter nenhum vértice de P em seu interior. Assim, podemos ligar b e p para produzir uma diagonal interna de P. A Figura 3 ilustra como localizar este vértice p. ◁

Demonstrações com o Uso da Propriedade da Boa Ordenação

A validade dos princípios da indução matemática e da indução completa é obtida a partir de um axioma fundamental do conjunto de números inteiros, a **propriedade da boa ordenação** (veja o Apêndice 1). Ela estabelece que todo conjunto não vazio de números inteiros não negativos tem pelo menos um menor elemento. Vamos mostrar como a propriedade da boa ordenação pode ser utilizada diretamente nas demonstrações. Além disso, podemos mostrar que ela, o princípio da indução matemática e a indução completa são todos equivalentes, ou seja, dada qualquer uma dessas três técnicas de demonstração, sua validade pode ser mostrada a partir da validade das outras duas técnicas. Na Seção 4.1 mostramos que o princípio da indução matemática é dado a partir da propriedade da boa ordenação. As outras partes dessa equivalência serão deixadas para os exercícios 31, 42 e 43, no final desta seção.

A PROPRIEDADE DA BOA ORDENAÇÃO Todo conjunto não vazio de números inteiros não negativos tem pelo menos um menor elemento.

A propriedade da boa ordenação pode ser usada diretamente em demonstrações.

EXEMPLO 5 Use a propriedade da boa ordenação para demonstrar o algoritmo de divisão. Lembre-se de que o algoritmo de divisão afirma que se a for um número inteiro e d for um número inteiro positivo, então existem números inteiros q e r com $0 \leq r < d$ e $a = dq + r$.

Exemplos Extras

Solução: Considere S como o conjunto de números inteiros não negativos na forma $a - dq$, em que q é um número inteiro. Este conjunto não é vazio, pois $-dq$ pode ser tão grande quanto desejar (considerando q como um número inteiro negativo com valor absoluto alto). Pela propriedade da boa ordenação, S tem um menor elemento $r = a - dq_0$.

O número inteiro r não é negativo. É também o caso de que $r < d$. Se ele não for, então terá um elemento não negativo menor em S, ou seja, $a - d(q_0 + 1)$. Para isso, suponha que $r \geq d$. Como $a = dq_0 + r$, temos que $a - d(q_0 + 1) = (a - dq_0) - d = r - d \geq 0$. Conseqüentemente, existem números inteiros q e r com $0 \leq r < d$. A demonstração de que q e r são únicos é deixada como exercício para o leitor. ◀

EXEMPLO 6 Em um torneio Round-robin, cada jogador joga com outro jogador apenas uma vez e cada jogada tem um vencedor e um perdedor. Dizemos que os jogadores $p_1, p_2, \ldots, p_m$ formam um *ciclo* se p_1 bater p_2, p_2 bater $p_3, \ldots, p_{m-1}$ bater p_m e p_m bater p_1. Use o princípio da boa ordenação para mostrar que se houver um ciclo de extensão m ($m \geq 3$) entre os jogadores em um torneio Round-robin, deverá haver um ciclo de três desses jogadores.

Solução: Assumimos que não haja ciclos de três jogadores. Como há pelo menos um ciclo no torneio todos contra todos, o conjunto de todos os números inteiros positivos n para o qual existe um ciclo de extensão n não é vazio. Pela propriedade da boa ordenação, este conjunto de números inteiros positivos tem pelo menos um elemento k, que por hipótese deve ser maior que três. Conseqüentemente, há um ciclo de jogadores $p_1, p_2, p_3, \ldots, p_k$ e não há ciclos menores.

Como não há ciclos de três jogadores, sabemos que $k > 3$. Considere os três primeiros elementos desse ciclo, p_1, p_2 e p_3. Há duas possibilidades de jogadas entre p_1 e p_3. Se p_3 bater p_1, temos que p_1, p_2, p_3 é um ciclo de extensão três, contradizendo nossa hipótese de que não há ciclos de três jogadores. Conseqüentemente, deverá haver o caso em que p_1 bate p_3. Isso significa que podemos omitir p_2 do ciclo $p_1, p_2, p_3, \ldots, p_k$ para obter o ciclo $p_1, p_3, p_4, \ldots, p_k$ de extensão $k - 1$, contradizendo a hipótese de que o menor ciclo tem extensão k. Concluímos que deve haver um ciclo de extensão três. ◀

Exercícios

1. Use a indução completa para mostrar que se você puder correr um ou dois quilômetros, e se uma vez que tenha corrido determinado número de quilômetros, puder sempre correr dois quilômetros a mais, então você pode correr qualquer número de quilômetros.
2. Use a indução completa para mostrar que todos os dominós caem em um arranjo infinito de dominós se soubermos que os três primeiros caem e que quando um dominó cai, aquele que fica três posições a frente também cai.
3. Considere $P(n)$ como a proposição que afirma que uma postagem de n centavos pode ser feita usando-se apenas selos de 3 e 5 centavos. Os itens deste exercício formam uma demonstração por indução completa de que $P(n)$ é verdadeira para $n \geq 8$.
   a) Mostre que as proposições $P(8)$, $P(9)$ e $P(10)$ são verdadeiras, completando o passo base da demonstração.
   b) Qual é a hipótese indutiva da demonstração?
   c) O que você precisa para demonstrar o passo de indução?
   d) Complete o passo de indução para $k \geq 10$.
   e) Explique por que esses passos mostram que esta proposição é verdadeira sempre que $n \geq 8$.
4. Considere $P(n)$ como a proposição que afirma que uma postagem de n centavos pode ser feita usando-se apenas selos de 4 e 7 centavos. Os itens deste exercício formam uma demonstração por indução completa de que $P(n)$ é verdadeira para $n \geq 18$.
   a) Mostre que as proposições $P(18)$, $P(19)$, $P(20)$ e $P(21)$ são verdadeiras, completando o passo base da demonstração.
   b) Qual é a hipótese indutiva da demonstração?
   c) O que você precisa para demonstrar o passo de indução?
   d) Complete a etapa indutiva para $k \geq 21$.
   e) Explique por que esses passos mostram que a proposição é verdadeira sempre que $n \geq 18$.
5. **a)** Determine quais postagens podem ser feitas usando-se apenas selos de 4 e 11 centavos.
   b) Demonstre sua resposta de (a) usando o princípio da indução matemática. Certifique-se de afirmar explicitamente sua hipótese indutiva no passo de indução.
   c) Demonstre sua resposta de (a) usando a indução completa. Em que a hipótese indutiva dessa demonstração difere da demonstração usada com indução matemática?
6. **a)** Determine quais postagens podem ser feitas usando-se apenas selos de 3 e 10 centavos.

b) Demonstre sua resposta de (a) usando o princípio da indução matemática. Certifique-se de afirmar explicitamente sua hipótese indutiva no passo de indução.

c) Demonstre sua resposta de (a) usando a indução completa. Em que difere a hipótese indutiva dessa demonstração da demonstração usada com indução matemática?

7. Qual a quantia de dinheiro que pode ser reunida usando apenas notas de \$ 2 e \$ 5? Demonstre sua resposta usando a indução completa.

8. Suponha que uma loja ofereça vales-presente nos valores de \$ 25 e \$ 40. Determine quais valores você pode juntar usando estes vales-presente. Demonstre sua resposta usando a indução completa.

***9.** Use a indução completa para demonstrar que $\sqrt{2}$ é um número irracional. [*Dica:* Considere $P(n)$ como a proposição de que $\sqrt{2} \neq n/b$ para qualquer número inteiro positivo b.]

10. Assuma que uma barra de chocolate tenha n quadrados organizados em formato retangular. A barra, com um pedaço retangular a menos que a barra original, pode ser quebrada na horizontal ou na vertical separando-se os quadrados. Admitindo que apenas um pedaço pode ser quebrado de cada vez, quantas vezes você deve quebrar a barra sucessivamente em n quadrados separados? Use a indução completa para demonstrar sua resposta.

11. Considere esta variação do jogo de Nim. O jogo começa com n jogadas. Dois jogadores podem remover as cartas uma, duas ou três de cada vez. O jogador que remover a última carta, perde. Use a indução completa para mostrar que se cada jogador jogar com a melhor estratégia possível, o primeiro vence, se $n = 4j$, $4j + 2$ ou $4j + 3$ para qualquer número inteiro não negativo j, e o segundo jogador vence no outro caso possível, quando $n = 4j + 1$ para qualquer número inteiro não negativo j.

12. Use a indução completa para mostrar que todo número inteiro positivo n pode ser escrito como uma soma de potências distintas de dois, ou seja, como uma soma de um subconjunto de números inteiros $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, e assim por diante. [*Dica:* Para o passo de indução, considere separadamente o caso em que $k + 1$ é par e em que ele é ímpar. Quando for par, note que $(k + 1)/2$ é um número inteiro.]

***13.** Um quebra-cabeça é montado por junções sucessivas de peças que se organizam em blocos. Um movimento é feito cada vez que uma peça é adicionada a um bloco, ou quando dois blocos são agrupados. Use a indução completa para demonstrar que não importa como os movimentos são realizados, são necessários $n - 1$ movimentos para montar um quebra-cabeça com n peças.

14. Suponha que você comece com uma pilha de n pedras e divida-a em n pilhas com uma pedra cada, separando, sucessivamente, uma pilha de pedras em duas menores. Cada vez que você faz a divisão, multiplica o número de pedras em cada uma das pilhas menores formadas; para que elas tenham r e s pedras, respectivamente, você computa rs. Mostre que não importa como você separa as pilhas, a soma dos produtos computados em cada etapa é igual a $n(n - 1)/2$.

15. Prove que o primeiro jogador tem uma estratégia vencedora para o jogo Chomp, introduzido no Exemplo 12 da Seção 1.7, se o quadro inicial for quadrado. [*Dica:* Use a indução completa para mostrar que esta estratégia funciona. Para o primeiro movimento, o primeiro jogador mastiga todos os biscoitos, exceto aqueles da esquerda e da margem superior. Nos movimentos subseqüentes, depois que o segundo jogador mastigar os biscoitos do topo ou da extrema esquerda, o primeiro jogador mastiga os biscoitos nas mesmas posições relativas da esquerda ou da margem superior, respectivamente.]

***16.** Demonstre que o primeiro jogador tem uma estratégia para ganhar o jogo Chomp, introduzido no Exemplo 12 da Seção 1.7, se o quadro inicial tiver dois quadrados de largura, ou seja, um quadro de $2 \times n$. [*Dica:* Use a indução completa. O primeiro movimento do primeiro jogador deve ser mastigar o biscoito na linha inferior da extremidade direita.]

17. Use a indução completa para mostrar que se um polígono simples com pelo menos quatro lados for triangulado, então pelo menos dois desses triângulos da triangulação têm dois lados que delimitam o exterior do polígono.

***18.** Use a indução completa para mostrar que quando um polígono convexo P com vértices consecutivos $v_1, v_2, \ldots, v_n$ é triangulado em $n - 2$ triângulos, os $n - 2$ triângulos podem ser numerados em $1, 2, \ldots, n - 2$ para que v_i seja um vértice do triângulo i, para $i = 1, 2, \ldots, n - 2$.

***19.** O **teorema de Pick** diz que a área de um polígono simples P no plano com vértices que são todos pontos reticulados (isto é, com coordenadas inteiras) é igual a $I(P) + B(P)/2 - 1$, em que $I(P)$ e $B(P)$ são os números de pontos reticulados no interior de P e na borda de P, respectivamente. Use a indução completa sobre o número de vértices de P para demonstrar o teorema de Pick. [*Dica:* Para o passo base, primeiro demonstre o teorema para retângulos, depois para triângulos retos e finalmente para todos os triângulos, notando que a área de um triângulo é a área de um retângulo que contém a área de no máximo três triângulos subtraída. Para o passo de indução, utilize o Lema 1.]

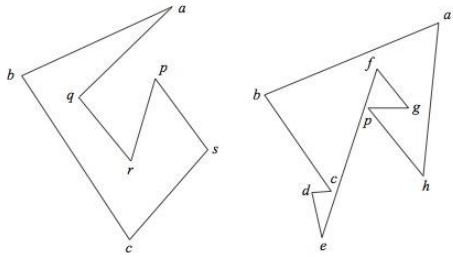
****20.** Suponha que P seja um polígono simples com vértices $v_1, v_2, \ldots, v_n$ listados como vértices consecutivos que estão ligados por um lado, e v_1 e v_n estão ligados por outro lado. Um vértice v_i é chamado de **orelha** se o segmento de reta que liga dois vértices adjacentes a v_i for uma diagonal interna do polígono simples. Duas orelhas v_i e v_j são chamadas de **não sobrepostas** se os interiores dos triângulos com vértices v_i e seus dois vértices adjacentes e v_j e seus dois vértices adjacentes não se cruzarem. Demonstre que todo polígono simples com pelo menos quatro vértices tem pelo menos duas orelhas não sobrepostas.

21. Na demonstração do Lema 1, mencionamos que muitos métodos incorretos para encontrar um vértice p, tal que o segmento de reta bp seja uma diagonal interna de P, têm sido publicados. Este exercício apresenta algumas maneiras incorretas p que foram escolhidas nessas demonstrações. Mostre, considerando um dos polígonos desenhados aqui, que para cada uma dessas escolhas de p, o segmento de reta bp não é necessariamente uma diagonal interna de P.

a) p é o vértice de P, tal que o ângulo $\angle abp$ é o menor.

b) p é o vértice de P com a menor coordenada x (diferente de b).

c) p é o vértice de P mais próximo de b.



Os exercícios 22 e 23 apresentam exemplos que mostram como a carga indutiva pode ser utilizada para demonstrar resultados em geometria computacional.

***22.** Considere $P(n)$ como a proposição que afirma que quando as diagonais que não se cruzam são desenhadas em um polígono convexo com n lados, pelo menos dois vértices do polígono não são pontos finais de qualquer uma dessas diagonais.

a) Mostre que quando tentamos demonstrar $P(n)$ para todos os números inteiros n com $n \geq 3$ usando a indução completa, o passo de indução não se sustenta.

b) Mostre que podemos demonstrar que $P(n)$ é verdadeira para todos os números inteiros n com $n \geq 3$, demonstrando pela indução completa a asserção forte $Q(n)$, para $n \geq 4$, em que $Q(n)$ afirma que sempre que diagonais que não se cruzam são desenhadas dentro de um polígono convexo com n lados, pelo menos dois vértices *não adjacentes* não são pontos finais de qualquer uma dessas diagonais.

23. Considere $E(n)$ como a proposição que afirma que em uma triangulação de um polígono simples de n lados, pelo menos um dos triângulos da triangulação tem dois lados que delimitam o exterior do polígono.

a) Explique onde uma demonstração que usa a indução completa de que $E(n)$ seja verdadeira para todos os números inteiros $n \geq 4$ é executada com dificuldade.

b) Mostre que podemos demonstrar que $E(n)$ é verdadeira para todos os números inteiros $n \geq 4$, demonstrando, pela indução completa, que a proposição forte $T(n)$ para todos os números inteiros $n \geq 4$, que afirma que em toda triangulação de um polígono simples, pelo menos dois triângulos da triangulação têm dois lados que delimitam o exterior do polígono.

***24.** Uma tarefa estável, definida no preâmbulo do Exercício 58 da Seção 3.1, é chamada de **ideal para os pretendentes** se não houver tarefas estáveis nas quais um pretendente é colocado em frente de uma pretendente de sua preferência na tarefa estável. Use a indução completa para mostrar que o algoritmo de aceitação produz uma tarefa estável que é ideal para os pretendentes.

25. Suponha que $P(n)$ seja uma função proposicional. Determine se para cada número inteiro positivo n, a proposição $P(n)$ deve ser verdadeira, e justifique sua resposta, se

a) $p(1)$ for verdadeira; para todos os números inteiros positivos n, se $P(n)$ for verdadeira, então $P(n+2)$ é verdadeira.

b) $P(1)$ e $P(2)$ forem verdadeiros; para todos os números inteiros positivos n, se $P(n)$ e $P(n+1)$ forem verdadeiras, então $P(n+2)$ é verdadeira.

c) $P(1)$ for verdadeira; para todos os números inteiros positivos n, se $P(n)$ for verdadeira, então $P(2n)$ é verdadeira.

d $P(1)$ for verdadeira; para todos os números inteiros positivos n, se $P(n)$ for verdadeira, então $P(n+1)$ é verdadeira.

26. Suponha que $P(n)$ seja uma função proposicional. Determine se para todo número inteiro não negativo n, a proposição $P(n)$ deve ser verdadeira se

a) $P(0)$ for verdadeira; para todos os números inteiros não negativos n, se $P(n)$ for verdadeira, então $P(n+2)$ é verdadeira.

b) $P(0)$ for verdadeira; para todos os números inteiros não negativos n, se $P(n)$ for verdadeira, então $P(n+3)$ é verdadeira.

c) $P(0)$ e $P(1)$ forem verdadeiras; para todos os números inteiros não negativos n, se $P(n)$ e $P(n+1)$ forem verdadeiras, então $P(n+2)$ é verdadeira.

d) $P(0)$ for verdadeira; para todos os números inteiros não negativos n, se $P(n)$ for verdadeira, então $P(n+2)$ e $P(n+3)$ são verdadeiras.

27. Mostre que, se a proposição $P(n)$ for verdadeira para infinitos números inteiros positivos n e $P(n+1) \rightarrow P(n)$ for verdadeira para todos os números inteiros positivos n, então $P(n)$ é verdadeira para todos os números inteiros positivos n.

☞**28.** Considere b como um número inteiro dado e j como um número inteiro positivo dado. Mostre que, se $P(b)$, $P(b+1)$, . . . , $P(b+j)$ forem verdadeiras e $[P(b) \wedge P(b+1) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$ for verdadeira para todo número inteiro positivo $k \geq b+j$, então $P(n)$ é verdadeira para todos os números inteiros n com $n \geq b$.

29. O que há de errado com esta "demonstração" por indução completa?

"Teorema" Para todo número inteiro não negativo n, $5n = 0$.

Passo Base: $5 \cdot 0 = 0$.

Passo de Indução: Suponha que $5j = 0$ para todos os números inteiros não negativos j com $0 \leq j \leq k$. Escreva $k+1 = i+j$, em que i e j são números naturais menores que $k+1$. Pela hipótese indutiva, $5(k+1) = 5(i+j) = 5i + 5j = 0 + 0 = 0$.

***30.** Encontre a falha na seguinte "demonstração" de que $a^n = 1$ para todos os números inteiros não negativos n, sempre que a for um número real diferente de zero.

Passo Base: $a^0 = 1$ é verdadeira pela definição de a^0.

Passo de Indução: Assuma que $a^j = 1$ para todos os números inteiros não negativos j com $j \leq k$. Então, note que

$$a^{k+1} = \frac{a^k \cdot a^k}{a^{k-1}} = \frac{1 \cdot 1}{1} = 1.$$

***31.** Mostre que a indução completa é um método de demonstração válido, indicando que ela se sustenta a partir da propriedade da boa ordenação.

32. Encontre a falha na seguinte "demonstração" de que toda postagem de três centavos ou mais pode ser feita usando-se apenas selos de três e quatro centavos.

Passo Base: Podemos fazer postagens de três centavos com apenas um selo de três centavos, e podemos fazer postagens de quatro centavos usando apenas um selo de quatro centavos.

Passo de Indução: Assuma que podemos fazer postagens de j centavos para todos os números inteiros não negativos j com $j \leq k$ usando apenas selos de três e quatro centavos. Então, podemos fazer postagens de $k+1$ centavos substi-

tuindo um selo de três centavos por um selo de quatro centavos ou substituindo dois selos de quatro centavos por três selos de três centavos.

33. Mostre que podemos demonstrar que $P(n, k)$ é verdadeira para todos os pares de números inteiros positivos n e k, se mostrarmos que

a) $P(1, 1)$ é verdadeira e $P(n, k) \rightarrow [P(n + 1, k) \wedge P(n, k + 1)]$ é verdadeira para todos os números inteiros positivos n e k.

b) $P(1, k)$ é verdadeira para todos os números inteiros positivos k, e $P(n, k) \rightarrow P(n + 1, k)$ é verdadeira para todos os números inteiros positivos n e k.

c) $P(n, 1)$ é verdadeira para todos os números inteiros positivos n, e $P(n, k) \rightarrow P(n, k + 1)$ é verdadeira para todos os números inteiros positivos n e k.

34. Demonstre que $\sum_{j=1}^{n} j(j + 1)(j + 2) \cdots (j + k - 1) = n(n + 1)(n + 2) \cdots (n + k)/(k + 1)$ para todos os números inteiros positivos k e n. [*Dica*: Use a técnica utilizada no Exercício 33.]

***35.** Mostre que, se $a_1, a_2, \ldots, a_n$ forem n números reais distintos, exatamente $n - 1$ multiplicações são utilizadas para computar os produtos desses n números, não importando quantos parênteses são inseridos em seus produtos. [*Dica:* Use a indução completa e considere a última multiplicação.]

***36.** A propriedade de boa ordenação pode ser utilizada para mostrar que há um único máximo divisor comum de dois números inteiros positivos. Considere a e b como números inteiros positivos, e considere S como o conjunto dos números inteiros positivos na forma $as + bt$, em que s e t são números inteiros.

a) Mostre que S não é vazio.

b) Use a propriedade da boa ordenação para mostrar que S tem um menor elemento c.

c) Mostre que se d for um divisor comum de a e b, então d é um divisor de c.

d) Mostre que $c \mid a$ e $c \mid b$. [*Dica:* Primeiro, assuma que $c \nmid a$. Então, $a = qc + r$, em que $0 < r < c$. Mostre que $r \in S$, contradizendo a escolha por c.]

e) Conclua a partir dos itens (c) e (d) que o máximo divisor comum de a e b existe. Termine a demonstração mostrando que este máximo divisor comum é único.

37. Considere a como um número inteiro e d como um número inteiro positivo. Mostre que os inteiros q e r com $a = dq + r$ e $0 \leq r < d$, que foram mostrados como existentes no Exemplo 5, são únicos.

38. Use a indução matemática para mostrar que um tabuleiro de damas retangular com um número par de células e dois quadrados faltando, um branco e um preto, pode ser preenchido por dominós.

****39.** Você pode usar a propriedade da boa ordenação para demonstrar a proposição: "Todo número inteiro positivo pode ser descrito usando não mais que quinze palavras em inglês"? Assuma que as palavras venham de determinado dicionário de língua inglesa. [*Dica:* Suponha que existam números inteiros positivos que não podem ser descritos usando não mais que quinze palavras em inglês. Pela boa ordenação, *o menor número inteiro positivo que não pode ser descrito usando não mais que quinze palavras em inglês* deveria então existir.]

40. Use a propriedade da boa ordenação para mostrar que se x e y forem números reais com $x < y$, então existe um número racional r com $x < r < y$. [*Dica:* Use a propriedade de Arquimedes, dada no Apêndice 1, para encontrar um número inteiro positivo A com $A > 1/(y - x)$. Então, mostre que existe um número racional r com denominador A entre x e y, procurando os números $\lfloor x \rfloor + j/A$, em que j é um número inteiro positivo.]

***41.** Mostre que a propriedade da boa ordenação pode ser demonstrada quando o princípio da indução matemática for tido como axioma.

***42.** Mostre que o princípio da indução matemática e a indução completa são equivalentes, ou seja, cada um pode ser mostrado como válido a partir do outro.

***43.** Mostre que podemos demonstrar a propriedade da boa ordenação quando tomamos o princípio da indução matemática ou o da indução completa como um axioma em vez de tomar a propriedade da boa ordenação como um axioma.

4.3 Definições Recursivas e Indução Estrutural

Introdução

Às vezes é difícil definir um objeto explicitamente. Entretanto, pode ser fácil defini-lo em termos dele próprio. Esse processo é chamado de **recursão.** Por exemplo, a ilustração mostrada na Figura 1 é produzida recursivamente. Primeiro, é dada uma ilustração. Então, é realizado um processo de sobreposições de sucessivas centralizações de fotos menores sobre a ilustração anterior.

Podemos usar a recursão para definir seqüências, funções e conjuntos. Nas discussões anteriores, especificamos os termos de uma seqüência que usa uma fórmula explícita. Por exemplo, a seqüência de potências de 2 é dada por $a_n = 2^n$ para $n = 0, 1, 2, \ldots$. Entretanto, esta seqüência pode também ser definida dando-se o primeiro termo da seqüência, ou seja, $a_0 = 1$ e uma regra para encontrar um termo a partir do anterior, ou seja, $a_{n+1} = 2a_n$ para $n = 0, 1, 2, \ldots$. Quando definimos uma seqüência *recursivamente*, especificando como os seus termos são

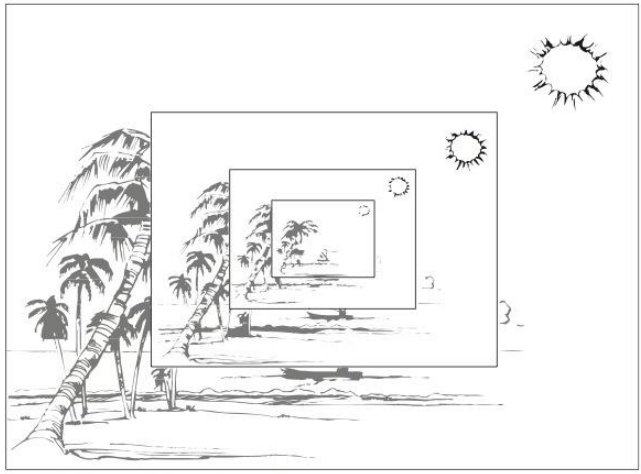


FIGURA 1 Uma Ilustração Definida Recursivamente.

encontrados a partir de termos anteriores, podemos usar a indução para demonstrar os resultados da seqüência.

Podemos definir um conjunto recursivamente, especificando alguns elementos iniciais em um passo base e fornecendo uma regra para a construção de novos elementos a partir daqueles obtidos no passo recursivo. Para demonstrar os resultados de recursividade em conjuntos, usamos um método chamado de *indução estrutural*, ou *recursão*.

Funções Definidas Recursivamente

Usamos duas etapas para definir uma função com o conjunto dos números inteiros não negativos como seu domínio:

PASSO BASE: Especifique o valor da função em zero.

Auto-avaliação

PASSO RECURSIVO: Forneça uma regra para encontrar seu valor em um número inteiro a partir dos valores nos números inteiros menores.

Essa definição é chamada de **recursividade** ou **definição indutiva.**

EXEMPLO 1 Suponha que f seja definida recursivamente por

Exemplos Extras

$$f(0) = 3,$$
$$f(n+1) = 2f(n) + 3.$$

Encontre $f(1)$, $f(2)$, $f(3)$ e $f(4)$.

Solução: A partir da definição recursiva, temos que

$$f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9,$$
$$f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21,$$
$$f(3) = 2f(2) + 3 = 2 \cdot 21 + 3 = 45,$$
$$f(4) = 2f(3) + 3 = 2 \cdot 45 + 3 = 93.$$ ◀

Muitas funções podem ser estudadas usando suas definições recursivas. A função fatorial é uma delas.

EXEMPLO 2 Dê uma definição recursiva da função fatorial $F(n) = n!$.

Solução: Podemos definir a função fatorial especificando seu valor inicial, ou seja $F(0) = 1$, e dando uma regra para encontrar $F(n + 1)$ a partir de $F(n)$. Isso é obtido notando que $(n + 1)!$ é computado a partir de $n!$ multiplicado por $n + 1$. Assim, a regra desejada é

$$F(n + 1) = (n + 1)F(n).$$ ◀

Para determinar um valor de uma função fatorial, tal como $F(5) = 5!$ a partir da definição recursiva encontrada no Exemplo 2, é necessário usar a regra que mostra como expressar $F(n + 1)$ em termos de $F(n)$ muitas vezes:

$$F(5) = 5F(4) = 5 \cdot 4F(3) = 5 \cdot 4 \cdot 3F(2) = 5 \cdot 4 \cdot 3 \cdot 2F(1)$$
$$= 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 \cdot F(0) = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 \cdot 1 = 120.$$

Uma vez que $F(0)$ é o único valor da função que aparece, não há necessidade de mais reduções. A única coisa que resta fazer é inserir o valor de $F(0)$ na fórmula.

As funções definidas recursivamente são **bem definidas**, ou seja, para todo número inteiro positivo, o valor da função neste inteiro é determinado de uma forma não ambígua. Isso significa que com qualquer número inteiro positivo, podemos usar as duas partes da definição para encontrar o valor da função naquele inteiro, e também que obtemos o mesmo valor, não importando como aplicamos as duas partes da definição. Isso é uma conseqüência do princípio da indução matemática. (Veja o Exercício 56 no final desta seção.) Outros exemplos de definições recursivas são dados nos exemplos 3 e 4.

EXEMPLO 3 Dê uma definição recursiva de a^n, em que a é um número real diferente de zero e n é um número inteiro não negativo.

Solução: A definição recursiva contém duas partes. Primeiro a^0 é determinado, ou seja, $a^0 = 1$. Então, é dada a regra pra encontrar a^{n+1} a partir de a^n, ou seja, $a^{n+1} = a \cdot a^n$, para $n = 0, 1, 2, 3, \ldots$. Estas duas equações definem unicamente a^n para todos os números inteiros não negativos n. ◀

EXEMPLO 4 Dê uma definição recursiva de

$$\sum_{k=0}^{n} a_k.$$

Solução: A primeira parte da definição recursiva é

$$\sum_{k=0}^{0} a_k = a_0.$$

A segunda parte é

$$\sum_{k=0}^{n+1} a_k = \left(\sum_{k=0}^{n} a_k\right) + a_{n+1}.$$

◀

Em algumas definições recursivas de funções, os valores da função dos primeiros k números inteiros positivos são especificados e uma regra é dada para determinar o valor da função para números inteiros maiores a partir de seus valores para alguns ou todos os k números inteiros precedentes. Essas definições recursivas são fixadas dessa maneira para produzir funções bem definidas a partir da indução completa (veja o Exercício 57 no final desta seção).

DEFINIÇÃO 1 Os *números de Fibonacci*, $f_0, f_1, f_2, \ldots$, são definidos pelas equações $f_0 = 0, f_1 = 1$ e

$$f_n = f_{n-1} + f_{n-2}$$

para $n = 2, 3, 4, \ldots$.

Links

EXEMPLO 5 Encontre os números de Fibonacci f_2, f_3, f_4, f_5 e f_6.

Solução: Como a primeira parte da definição afirma que $f_0 = 0$ e $f_1 = 1$, temos, a partir da segunda parte da definição, que

$$f_2 = f_1 + f_0 = 1 + 0 = 1,$$
$$f_3 = f_2 + f_1 = 1 + 1 = 2,$$
$$f_4 = f_3 + f_2 = 2 + 1 = 3,$$
$$f_5 = f_4 + f_3 = 3 + 2 = 5,$$
$$f_6 = f_5 + f_4 = 5 + 3 = 8.$$

◀

Podemos usar a definição recursiva dos números de Fibonacci para demonstrar muitas propriedades desses números. Daremos uma dessas propriedades no Exemplo 6.

EXEMPLO 6 Mostre que, sempre que $n \geq 3, f_n > \alpha^{n-2}$, em que $\alpha = (1 + \sqrt{5})/2$.

Exemplos Extras

Solução: Podemos usar a indução completa para demonstrar esta inequação. Considere $P(n)$ como a proposição de que $f_n > \alpha^{n-2}$. Queremos mostrar que $P(n)$ é verdadeira sempre que n for um número inteiro maior que ou igual a 3.

PASSO BASE: Primeiro, note que

$$\alpha < 2 = f_3, \qquad \alpha^2 = (3 + \sqrt{5})/2 < 3 = f_4,$$

então, $P(3)$ e $P(4)$ são verdadeiras.

PASSO DE INDUÇÃO: Assuma que $P(j)$ seja verdadeira, ou seja, que $f_j > \alpha^{j-2}$, para todos os números inteiros j com $3 \le j \le k$, em que $k \ge 4$. Devemos mostrar que $P(k+1)$ é verdadeira, ou seja, que $f_{k+1} > \alpha^{k-1}$. Como α é uma solução de $x^2 - x - 1 = 0$ (como a fórmula quadrática mostra), temos que $\alpha^2 = \alpha + 1$. Assim,

$$\alpha^{k-1} = \alpha^2 \cdot \alpha^{k-3} = (\alpha + 1)\,\alpha^{k-3} = \alpha \cdot \alpha^{k-3} + 1 \cdot \alpha^{k-3} = \alpha^{k-2} + \alpha^{k-3}.$$

Pela hipótese indutiva, se $k \ge 4$, temos que

$$f_{k-1} > \alpha^{k-3}, \qquad f_k > \alpha^{k-2}.$$

Assim, temos

$$f_{k+1} = f_k + f_{k-1} > \alpha^{k-2} + \alpha^{k-3} = \alpha^{k-1}.$$

Temos que $P(k+1)$ é verdadeira. Isso completa a demonstração. ◀

Lembre-se: O passo de indução mostra que, sempre que $k \ge 4$, $P(k+1)$ é dado a partir da hipótese de que $P(j)$ seja verdadeira para $3 \le j \le k$. Assim, esse passo *não* mostra que $P(3) \rightarrow P(4)$. Além disso, mostramos que $P(4)$ é verdadeira separadamente.

Podemos agora mostrar que o algoritmo de Euclides usa $O(\log b)$ divisões para encontrar o máximo divisor comum dos números inteiros positivos a e b, em que $a \ge b$.

TEOREMA 1

TEOREMA DE LAMÉ Considere a e b como números inteiros positivos com $a \ge b$. Então, o número de divisões utilizado pelo algoritmo de Euclides para encontrar mdc(a, b) é menor que ou igual a cinco vezes o número de dígitos decimais em b.

Demonstração: Lembre-se de que, quando o algoritmo de Euclides é aplicado para encontrar o mdc(a, b) com $a \ge b$, esta seqüência de equações (em que $a = r_0$ e $b = r_1$) é obtida.

$$\begin{aligned}
r_0 &= r_1 q_1 + r_2 && 0 \le r_2 < r_1 \\
r_1 &= r_2 q_2 + r_3 && 0 \le r_3 < r_2 \\
&\vdots \\
r_{n-2} &= r_{n-1} q_{n-1} + r_n && 0 \le r_n < r_{n-1} \\
r_{n-1} &= r_n q_n.
\end{aligned}$$

Links

Aqui, foram utilizadas n divisões para encontrar $r_n = \text{mdc}(a, b)$. Note que os quocientes $q_1, q_2, \ldots, q_{n-1}$ são todos pelo menos 1. Além disso, $q_n \ge 2$, porque $r_n < r_{n-1}$. Isso implica que

FIBONACCI (1170–1250) Fibonacci (abreviação de *filius Bonacci*, ou "filho de Bonacci") foi também conhecido como Leonardo de Pisa. Ele nasceu no centro comercial italiano de Pisa e foi um comerciante que viajava muito para o Oriente Médio, onde fez contato com matemáticos árabes. Em seu livro, *Liber Abaci*, Fibonacci introduziu ao mundo europeu a notação arábica para numerais e algoritmos na aritmética. Foi em seu livro que seu famoso problema dos coelhos (descrito na Seção 7.1) apareceu. Fibonacci também escreveu livros sobre geometria e trigonometria e sobre equações diofantinas, que basicamente procuram soluções inteiras para as equações.

$$
\begin{aligned}
r_n &\geq 1 = f_2,\\
r_{n-1} &\geq 2r_n \geq 2f_2 = f_3,\\
r_{n-2} &\geq r_{n-1} + r_n \geq f_3 + f_2 = f_4,\\
&\;\;\vdots\\
r_2 &\geq r_3 + r_4 \geq f_{n-1} + f_{n-2} = f_n,\\
b &= r_1 \geq r_2 + r_3 \geq f_n + f_{n-1} = f_{n+1}.
\end{aligned}
$$

Temos que, se n divisões forem usadas pelo algoritmo de Euclides para encontrar mdc(a, b) com $a \geq b$, então $b \geq f_{n+1}$. A partir do Exemplo 6, sabemos que $f_{n+1} > \alpha^{n-1}$ para $n > 2$, em que $\alpha = (1+\sqrt{5})/2$. Assim, temos que $b > \alpha^{n-1}$. Além disso, como $\log_{10} \alpha \sim 0{,}208 > 1/5$, vemos que

$$\log_{10} b > (n-1)\log_{10} \alpha > (n-1)/5.$$

Assim, $n - 1 < 5 \cdot \log_{10} b$. Agora suponha que b tenha k dígitos decimais. Então $b < 10^k$ e $\log_{10} b < k$. Temos que $n - 1 < 5k$, e como k é um número inteiro, temos que $n \leq 5k$. Isso finaliza a demonstração. ◁

Como o número de dígitos decimais de b é igual a $\lfloor \log_{10} b \rfloor + 1$, que é menor que ou igual a $\log_{10} b + 1$, o Teorema 1 nos diz que o número de divisões necessárias para encontrar o mdc(a, b) com $a > b$ é menor que ou igual a $5(\log_{10} b + 1)$. Como $5(\log_{10} b + 1)$ é $O(\log b)$, vemos que $O(\log b)$ divisões são usadas pelo algoritmo de Euclides para encontrar o mdc(a, b) sempre que $a > b$.

Conjuntos e Estruturas Definidas Recursivamente

Auto-avaliação

Exploramos como as funções podem ser definidas recursivamente. Agora veremos como os conjuntos podem ser definidos desse mesmo modo. Como na definição recursiva de funções, as definições recursivas de conjuntos têm dois passos, um **passo base** e um **passo recursivo.** No passo base, o grupo inicial de elementos é determinado. No passo recursivo, são fornecidas as regras para formar novos elementos no conjunto a partir daqueles que já são conhecidos. As definições recursivas podem incluir também uma **regra de exclusão**, que especifica que um conjunto definido recursivamente contém não mais que aqueles elementos especificados no passo base ou construídos pelas aplicações do passo recursivo. Em nossas discussões, sempre assumiremos tacitamente que a regra de exclusão é válida e nenhum elemento pertence ao conjunto definido recursivamente, a menos que esteja no grupo inicial especificado no passo base ou que possa ser

Links

GABRIEL LAMÉ (1795–1870) Gabriel Lamé entrou para a Escola Politécnica, na França, em 1813, graduando-se em 1817. Ele continuou sua educação na École des Mines, graduando-se em 1820.

Em 1820, Lamé foi à Rússia, onde trabalhou como diretor das Escolas de Estradas e Transporte em São Petersburgo. Não apenas ensinava, como também projetava estradas e pontes na Rússia. Ele retornou a Paris em 1832, onde ajudou a fundar uma empresa de engenharia. Entretanto, logo deixou a empresa, aceitando a cadeira de física na Escola Politécnica, onde permaneceu até 1844. Enquanto estava nesse cargo, ele atuava fora da academia como consultor em engenharia, trabalhando como engenheiro-chefe de minas e participando de construções de estradas.

A contribuição inicial de Lamé se deu em teoria dos números, matemática aplicada e termodinâmica. Seu trabalho mais conhecido envolve a introdução de coordenadas curvas. Em teoria dos números, seu trabalho inclui demonstrar o Último Teorema de Fermat para $n = 7$, assim como fornecer um limite superior para o número de divisões usado pelo algoritmo de Euclides dado neste texto. Na opinião de Gauss, um dos mais importantes matemáticos da época, Lamé foi o matemático francês que mais avançou em seus estudos naquela época. Entretanto, os matemáticos franceses consideravam-no muito prático, enquanto os cientistas franceses consideravam-no muito teórico.

gerado usando-se o passo recursivo uma ou mais vezes. Depois, veremos como podemos usar uma técnica conhecida como indução estrutural para demonstrar os resultados de conjuntos definidos recursivamente.

Os exemplos 7, 8, 10 e 11 ilustram a definição recursiva de conjuntos. Em cada exemplo, mostramos os elementos gerados pelas primeiras aplicações do passo recursivo.

EXEMPLO 7 Considere o subconjunto S do conjunto dos números inteiros definido por

PASSO BASE: $3 \in S$.

PASSO RECURSIVO: Se $x \in S$ e $y \in S$, então $x + y \in S$.

Exemplos Extras

Os novos elementos encontrados em S são 3, pelo passo base, $3 + 3 = 6$, na primeira aplicação do passo recursivo, $3 + 6 = 6 + 3 = 9$ e $6 + 6 = 12$, na segunda aplicação do passo recursivo, e assim por diante. Mostraremos depois que S é o conjunto dos múltiplos positivos de 3. ◀

As definições recursivas desempenham importante papel no estudo de cadeias. (Veja o Capítulo 12 para uma introdução da teoria das linguagens formais, por exemplo.) Lembre-se da Seção 2.4, em que uma cadeia de um alfabeto Σ é uma seqüência finita de símbolos de Σ. Podemos definir Σ^*, o conjunto das cadeias sobre Σ, recursivamente, como mostra a Definição 2.

DEFINIÇÃO 2 O conjunto Σ^* de *cadeias* do alfabeto Σ pode ser definido recursivamente por

PASSO BASE: $\lambda \in \Sigma^*$ (em que λ é a cadeia vazia que não contém símbolos).

PASSO RECURSIVO: se $w \in \Sigma^*$ e $x \in \Sigma$, então $wx \in \Sigma^*$.

O passo base da definição recursiva de cadeias diz que a cadeia vazia pertence a Σ^*. O passo recursivo afirma que novas cadeias são produzidas pela adição de um símbolo a partir de Σ ao final das cadeias em Σ^*. Em cada aplicação do passo recursivo, cadeias com um símbolo adicional são geradas.

EXEMPLO 8 Se $\Sigma = \{0, 1\}$, as cadeias encontradas que estão em Σ^*, o conjunto de todas as cadeias, são λ, especificadas para pertencer a Σ^* no passo base, 0 e 1, formados durante a primeira aplicação do passo recursivo, 00, 01, 10 e 11, formados durante a segunda aplicação do passo recursivo, e assim por diante. ◀

As definições recursivas podem ser usadas para operações ou funções de elementos de conjuntos definidos recursivamente. Isso é ilustrado na Definição 3 da concatenação de duas cadeias e no Exemplo 9 sobre a extensão de uma cadeia.

DEFINIÇÃO 3 Duas cadeias podem ser combinadas por meio de uma operação de *concatenação*. Considere Σ como o conjunto de símbolos e Σ^* como o conjunto de cadeias formadas a partir dos símbolos em Σ. Podemos definir a concatenação de duas cadeias, indicadas por $\cdot$, recursivamente, como ilustrado a seguir.

PASSO BASE: Se $w \in \Sigma^*$, então $w \cdot \lambda = w$, em que λ é a cadeia vazia.

PASSO RECURSIVO: Se $w_1 \in \Sigma^*$ e $w_2 \in \Sigma^*$ e $x \in \Sigma$, então $w_1 \cdot (w_2 x) = (w_1 \cdot w_2)x$.

A concatenação das cadeias w_1 e w_2 é geralmente escrita como $w_1 w_2$, em vez de $w_1 \cdot w_2$. Repetindo as aplicações do passo recursivo, temos que a concatenação de duas cadeias w_1 e w_2 consiste

em símbolos em w_1 seguidos de símbolos em w_2. Por exemplo, a concatenação de $w_1 = abra$ e $w_2 = cadabra$ é $w_1w_2 = abracadabra$.

EXEMPLO 9 **Comprimento de uma Cadeia** Dê uma definição recursiva de $l(w)$, o comprimento, ou a extensão, da cadeia w.

Solução: A extensão de uma cadeia pode ser definida por

$l(\lambda) = 0;$
$l(wx) = l(w) + 1$ se $w \in \Sigma^*$ e $x \in \Sigma$. ◀

Outro uso importante das definições recursivas é definir **fórmulas bem formadas** de vários tipos. Isso é ilustrado nos exemplos 10 e 11.

EXEMPLO 10 **Fórmulas Bem Formadas para Formas de Proposições Compostas** Podemos definir o conjunto das fórmulas bem formadas para formas de proposições compostas que envolvem **V**, **F**, variáveis proposicionais e operadores do conjunto $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$.

PASSO BASE: **V**, **F** e s, em que s é uma variável proposicional, são fórmulas bem formadas.

PASSO RECURSIVO: Se E e F são fórmulas bem formadas, então $(\neg E)$, $(E \wedge F)$, $(E \vee F)$, $(E \rightarrow F)$ e $(E \leftrightarrow F)$ são fórmulas bem formadas.

Por exemplo, pelo passo base, sabemos que **V**, **F**, p e q são fórmulas bem formadas, em que p e q são variáveis proposicionais. A partir de uma aplicação inicial do passo recursivo, sabemos que $(P \vee q)$, $(P \rightarrow \mathbf{F})$, $(\mathbf{F} \rightarrow q)$ e $(q \wedge \mathbf{F})$ são fórmulas bem formadas. Uma segunda aplicação do passo recursivo mostra que $((p \vee q) \rightarrow (q \wedge \mathbf{F}))$, $(q \vee (p \vee q))$ e $((p \rightarrow \mathbf{F}) \rightarrow \mathbf{V})$ são fórmulas bem formadas. Deixamos para o leitor mostrar que $p\neg \wedge q$, $pq \wedge$ e $\neg \wedge pq$ *não* são fórmulas bem formadas, mostrando que nenhuma delas pode ser obtida usando-se o passo base ou qualquer aplicação do passo recursivo. ◀

EXEMPLO 11 **Fórmulas Bem Formadas de Operações** Podemos definir o conjunto de fórmulas bem formadas composto de variáveis, numerais e operadores do conjunto $\{+, -, *, /, \uparrow\}$ (em que * indica multiplicação e $\uparrow$ indica exponenciação) recursivamente.

PASSO BASE: x é uma formula bem formada, se x for um numeral ou variável.

PASSO RECURSIVO: Se F e G forem fórmulas bem formadas, então $(F + G)$, $(F - G)$, $(F * G)$, (F/G) e $(F \uparrow G)$ são fórmulas bem formadas.

Por exemplo, pelo passo base, vemos que x, y, 0 e 3 são fórmulas bem formadas (assim como qualquer variável ou numeral). Fórmulas bem formadas construídas a partir da aplicação do passo recursivo uma vez incluem $(x + 3)$, $(3 + y)$, $(x - y)$, $(3 - 0)$, $(x * 3)$, $(3 * y)$, $(3/0)$, (x/y), $(3 \uparrow x)$ e $(0 \uparrow 3)$. Aplicando o passo recursivo duas vezes, vemos que as fórmulas como $((x + 3) + 3)$ e $(x - (3 * y))$ são bem formadas. [Note que $(3/0)$ é uma fórmula bem formada porque queremos que apenas a sintaxe seja relevante aqui.] Deixamos para o leitor mostrar que cada uma das fórmulas $x3 +$, $y * + x$ e $* x/y$ *não* são fórmulas bem formadas, mostrando que nenhuma delas pode ser obtida usando o passo base ou qualquer aplicação do passo recursivo. ◀

Estudaremos as árvores exaustivamente no Capítulo 10. Uma árvore é um tipo especial de grafo; um grafo é feito de vértices e arestas que conectam alguns pares de vértices. Estudaremos os grafos no Capítulo 9, mas os introduziremos brevemente aqui para ilustrar como podem ser definidos recursivamente.

DEFINIÇÃO 4 O conjunto de *árvores com raiz*, em que uma árvore com raiz consiste em um conjunto de vértices que contém um vértice distinto, chamado de *raiz*, e arestas que conectam esses vértices, pode ser definido recursivamente por estes passos:

PASSO BASE: Um único vértice r é uma árvore com raiz.

PASSO RECURSIVO: Suponha que $T_1, T_2, \ldots, T_n$ são árvores com raízes disjuntas que possuem raízes $r_1, r_2, \ldots, r_n$, respectivamente. Então, o grafo formado começando com uma raiz r, que não esteja em nenhuma das árvores com raízes $T_1, T_2, \ldots, T_n$, e adicionando uma aresta a partir de r a cada um dos vértices $r_1, r_2, \ldots, r_n$, é também uma árvore com raiz.

Na Figura 2 ilustramos as árvores com raízes formadas com o passo base e aplicando-se o passo recursivo uma e duas vezes. Note que infinitas árvores com raízes são formadas em cada aplicação da definição recursiva.

Árvores binárias são um tipo especial de árvores. Forneceremos as definições recursivas de dois tipos de árvores binárias — árvores binárias completas e árvores binárias estendidas. No passo recursivo da definição de cada tipo de árvore binária, duas árvores binárias são combinadas para formar uma nova árvore com uma dessas árvores colocada como subárvore à esquerda e a outra colocada como subárvore à direita. Nas árvores binárias estendidas, as subárvores à esquerda ou à direita podem ser vazias, mas, em árvores binárias completas, isso não é possível. Árvores binárias são um dos mais importantes tipos de estruturas em ciência da computação. No Capítulo 10 veremos de que maneira elas podem ser usadas nos algoritmos de busca e de ordenação, em algoritmos para comprimir dados e em muitas outras aplicações. Primeiro, definiremos as árvores binárias estendidas.

DEFINIÇÃO 5 O conjunto de *árvores binárias estendidas* pode ser definido recursivamente pelos passos abaixo:

PASSO BASE: O conjunto vazio é uma árvore binária estendida.

PASSO RECURSIVO: Se T_1 e T_2 forem árvores binárias estendidas disjuntas, haverá uma árvore binária estendida, indicada por $T_1 \cdot T_2$, que consiste em uma raiz r junto com arestas que conectam a raiz a cada uma das raízes da subárvore à esquerda T_1 e da subárvore à direita T_2 quando essas árvores não forem vazias.

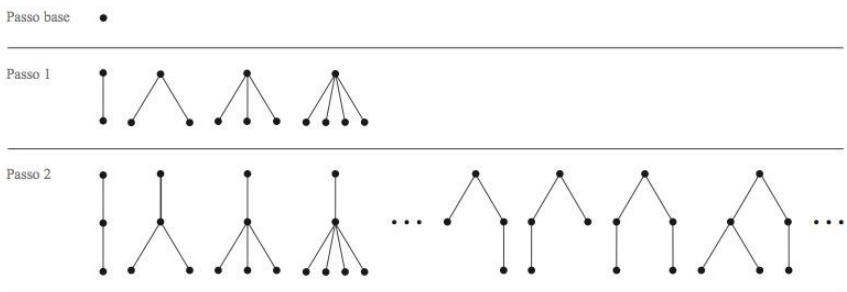


FIGURA 2 Construindo Árvores com Raízes.

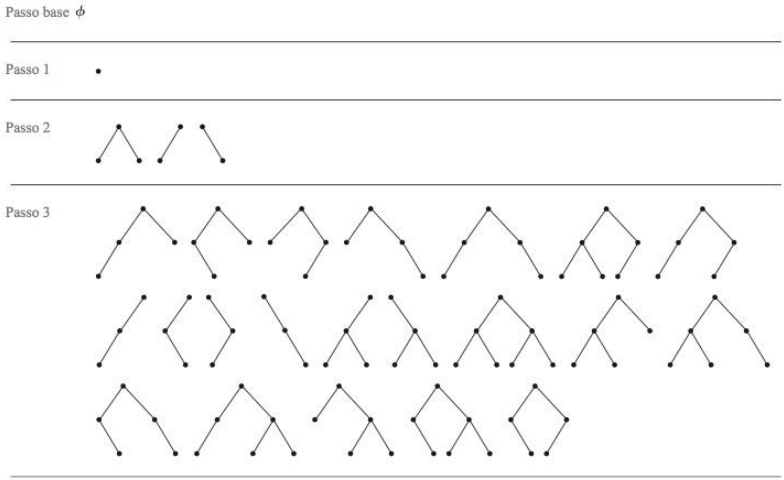


FIGURA 3 Construindo Árvores Binárias Estendidas.

A Figura 3 mostra como árvores binárias estendidas são construídas aplicando-se o passo recursivo de uma a três vezes.

Mostraremos agora como definir o conjunto das árvores binárias completas. Note que a diferença entre esta definição recursiva e a das árvores binárias estendidas fundamenta-se inteiramente no passo base.

DEFINIÇÃO 6 O conjunto das árvores binárias completas pode ser definido recursivamente pelos passos abaixo:

PASSO BASE: Há uma árvore binária completa que contém apenas um único vértice r.

PASSO RECURSIVO: Se T_1 e T_2 forem árvores binárias completas disjuntas, haverá uma árvore binária completa, indicada por $T_1 \cdot T_2$, que é formada por uma raiz r junto com as arestas que ligam a raiz a cada uma das raízes das subárvores à esquerda T_1 e à direita T_2.

A Figura 4 mostra como as árvores binárias completas são construídas aplicando-se o passo recursivo uma e duas vezes.

Indução Estrutural

Para demonstrar os resultados sobre conjuntos definidos recursivamente, geralmente usamos formas da indução matemática. O Exemplo 12 ilustra a conexão entre conjuntos definidos recursivamente e a indução matemática.

EXEMPLO 12 Mostre que o conjunto S definido no Exemplo 7, ao especificar que $3 \in S$ e que se $x \in S$ e $y \in S$, então $x + y \in S$, é o conjunto de todos os números inteiros positivos que são múltiplos de 3.

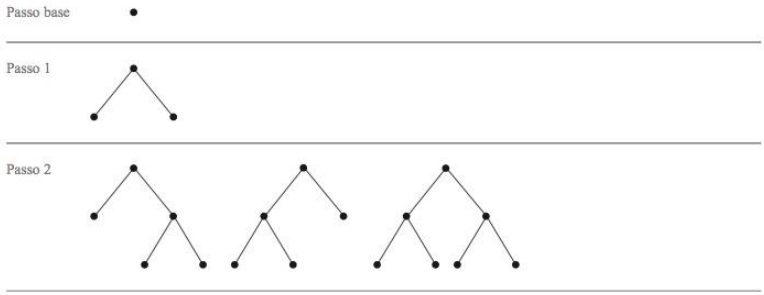


FIGURA 4 Construindo Árvores Binárias Completas.

Solução: Considere A como o conjunto de todos os números inteiros positivos divisíveis por 3. Para demonstrar que $A = S$, devemos mostrar que A é um subconjunto de S e que S é um subconjunto de A. Para demonstrar que A é um subconjunto de S, devemos mostrar que todo número inteiro positivo divisível por 3 está em S. Usaremos a indução matemática para demonstrar isso.

Considere $P(n)$ como a proposição de que $3n$ pertence a S. O passo base é mantido, pois, pela primeira parte da definição recursiva de S, $3 \cdot 1 = 3$ está em S. Para estabelecer o passo de indução, assuma que $P(k)$ seja verdadeira, ou seja, que $3k$ está em S. Como $3k$ está em S e como 3 está em S, temos que, a partir da segunda parte da definição recursiva de S, $3k + 3 = 3(k + 1)$ está também em S.

Para demonstrar que S é um subconjunto de A, usamos a definição recursiva de S. Primeiro, o passo base da definição especifica que 3 está em S. Como $3 = 3 \cdot 1$, todos os elementos considerados como estando em S neste passo são divisíveis por 3 e, portanto, estão em A. Para finalizar a demonstração, devemos mostrar que todos os números inteiros em S gerados por meio da segunda parte da definição recursiva estão em A. Isso consiste em mostrar que $x + y$ está em A sempre que x e y forem elementos de S também considerados como estando em A. Agora, se x e y estiverem em A, temos que $3 \mid x$ e $3 \mid y$. Pelo item (i) do Teorema 1 da Seção 3.4, temos que $3 \mid x + y$, completam a demonstração. ◀

No Exemplo 12 usamos a indução matemática sobre o conjunto dos números inteiros positivos e uma definição recursiva para demonstrar um resultado de conjuntos definidos recursivamente. Entretanto, em vez de usar a indução matemática diretamente para demonstrar resultados sobre conjuntos definidos recursivamente, podemos usar uma forma mais conveniente de indução, conhecida como **indução estrutural.** Uma demonstração por indução estrutural é formada por dois passos, que são:

PASSO BASE: Mostre como os resultados se mantêm para todos os elementos especificados no passo base da definição recursiva de pertencerem a um conjunto.

PASSO RECURSIVO: Mostre que, se a proposição for verdadeira para cada um dos elementos usados para formar novos elementos no passo recursivo da definição, o resultado se mantém para novos elementos.

A validade da indução estrutural ocorre a partir do princípio da indução matemática para números inteiros não negativos. Para verificar isso, considere $P(n)$ como a afirmação verdadeira para todos os elementos do conjunto que são gerados por n ou menos aplicações das regras no passo recursivo de uma definição recursiva. Teremos estabelecido que o princípio da indução matemática implica o princípio da indução estrutural se pudermos mostrar que $P(n)$ é verdadeira sempre que n for um número inteiro positivo. No passo base de uma demonstração por indução estrutural, mostramos que $P(0)$ é verdadeira, ou seja, mostramos que o resultado é verdadeiro para todos os elementos especificados como sendo do conjunto no passo base da definição. Uma conseqüência do passo de indução é que, se

assumimos que $P(k)$ seja verdadeira, temos que $P(k+1)$ é verdadeira. Quando completamos uma demonstração usando a indução estrutural, mostramos que $P(0)$ é verdadeira e que $P(k)$ implica $P(k+1)$. Pela indução matemática, temos que $P(n)$ é verdadeira para todos os números inteiros não negativos n. Isso também mostra que o resultado é verdadeira para todos os elementos gerados na definição recursiva e mostra que a indução estrutural é uma técnica válida de demonstração.

EXEMPLOS DE DEMONSTRAÇÕES QUE USAM A INDUÇÃO ESTRUTURAL A indução estrutural pode ser usada para demonstrar que todos os membros de um conjunto construídos recursivamente têm uma propriedade particular. Ilustraremos essa idéia usando a indução estrutural para demonstrar os resultados de fórmulas bem formadas, cadeias e árvores binárias. Para cada demonstração, teremos de tomar cuidado com os passos base e recursivo apropriados. Por exemplo, usar a indução estrutural para demonstrar um resultado sobre o conjunto de fórmulas bem formadas definidas no Exemplo 10, em que especificamos que **V**, **F** e toda variável proposicional s são fórmulas bem formadas e especificamos que, se E e F forem fórmulas bem formadas, então $(\neg E)$, $(E \wedge F)$, $(E \vee F)$, $(E \rightarrow F)$ e $(E \leftrightarrow F)$ são também fórmulas bem formadas, precisamos completar o passo base e o recursivo abaixo.

PASSO BASE: Mostre que o resultado é verdadeiro para **V**, **F** e s sempre que s for uma variável proposicional.

PASSO RECURSIVO: Mostre que, se o resultado for verdadeiro para as proposições compostas p e q, também será verdadeiro para $(\neg p)$, $(p \vee q)$, $(p \wedge q)$, $(p \rightarrow q)$ e $(p \leftrightarrow q)$.

O Exemplo 13 ilustra como podemos demonstrar os resultados de fórmulas bem formadas usando a indução estrutural.

EXEMPLO 13 Mostre que todas as fórmulas bem formadas para proposições compostas, como definidas no Exemplo 10, contêm um número igual de parênteses do lado esquerdo e do lado direito.

Solução:

PASSO BASE: Cada uma das formulas **V**, **F** e s não contém parênteses, assim, elas contêm um número igual de parênteses do lado esquerdo e do lado direito.

PASSO RECURSIVO: Assuma p e q como fórmulas bem formadas, cada uma com um número igual de parênteses à esquerda e à direita, ou seja, se l_P e l_q forem o número de parênteses à esquerda de p e q, respectivamente, e r_P e r_q forem o número de parênteses à direita de p e q, respectivamente, então $l_p = r_p$ e $l_q = r_q$. Para completar o passo de indução, precisamos mostrar que cada um de $(\neg p)$, $(p \vee q)$, $(p \wedge q)$, $(p \rightarrow q)$ e $(p \leftrightarrow q)$ também contém um número igual de parênteses à esquerda e à direita. O número de parênteses à esquerda na primeira dessas proposições compostas é igual a $l_p + 1$ e em cada uma das outras proposições compostas é igual a $l_p + l_q + 1$. Do mesmo modo, o número de parênteses à direita na primeira dessas proposições compostas é igual a $r_p + 1$ e em cada uma das outras proposições é igual a $r_p + r_q + 1$. Como $l_p = r_p$ e $l_q = r_q$, temos que cada uma das expressões compostas contém o mesmo número de parênteses à direita e à esquerda. Isso completa a demonstração indutiva. ◀

Suponha que $P(w)$ é uma função proposicional do conjunto de cadeias $w \in \Sigma^*$. Para usar a indução estrutural a fim de demonstrar que $P(w)$ é mantida para todas as cadeias $w \in \Sigma^*$, precisamos completar o passo base e o passo recursivo. Esses passos são:

PASSO BASE: Mostre que $P(\lambda)$ é verdadeira.

PASSO RECURSIVO: Assuma que $P(w)$ seja verdadeira, em que $w \in \Sigma^*$. Mostre que, se $x \in \Sigma$, então $P(wx)$ deve ser também verdadeira.

O Exemplo 14 ilustra como a indução estrutural pode ser usada em demonstrações de cadeias.

EXEMPLO 14 Use a indução estrutural para demonstrar que $l(xy) = l(x) + l(y)$, em que x e y pertencem a Σ^*, o conjunto de cadeias do alfabeto Σ.

Solução: Fundamentaremos nossa demonstração na definição recursiva do conjunto Σ^*, dado na Definição 2, e na definição de extensão de uma cadeia do Exemplo 9, que especifica que $l(\lambda) = 0$ e $l(wx) = l(w) + 1$ quando $w \in \Sigma^*$ e $x \in \Sigma$. Considere $P(y)$ como a proposição que afirma que $l(xy) = l(x) + l(y)$ sempre que x pertencer a Σ^*.

PASSO BASE: Para completar o passo base, devemos mostrar que $P(\lambda)$ é verdadeira, ou seja, devemos mostrar que $l(x\lambda) = l(x) + l(\lambda)$ para todo $x \in \Sigma^*$. Como $l(x\lambda) = l(x) = l(x) + 0 = l(x) + l(\lambda)$ para toda cadeia x, temos que $P(\lambda)$ é verdadeira.

PASSO RECURSIVO: Para completar o passo de indução, assumimos que $P(y)$ seja verdadeira e mostramos como isso implica que $P(ya)$ seja verdadeira sempre que $a \in \Sigma$. O que nós precisamos fazer é mostrar que $l(xya) = l(x) + l(ya)$ para todo $a \in \Sigma$. Para mostrar isso, note que, pela definição de recursividade de $l(w)$ (dada no Exemplo 9), temos que $l(xya) = l(xy) + 1$ e $l(ya) = l(y) + 1$. E, pela hipótese indutiva, $l(xy) = l(x) + l(y)$. Concluímos que $l(xya) = l(x) + l(y) + 1 = l(x) + l(ya)$. ◀

Podemos demonstrar os resultados sobre árvores ou classes especiais de árvores usando a indução estrutural. Por exemplo, para demonstrar um resultado de árvores binárias completas usando a indução estrutural, precisamos completar o passo base e o passo recursivo a seguir.

PASSO BASE: Mostre que o resultado é verdadeiro para a árvore que tiver um único vértice.

PASSO RECURSIVO: Mostre que se o resultado for verdadeiro para as árvores T_1 e T_2, então é verdadeiro para a árvore $T_1 \cdot T_2$ que possui uma raiz r, e que tem a subárvore T_1 à sua esquerda e a subárvore T_2 à sua direita.

Antes de fornecermos um exemplo que mostra como a indução estrutural pode ser usada para demonstrar um resultado de árvores binárias completas, precisamos de algumas definições. Definiremos recursivamente a altura $h(T)$ e o número de vértices $n(T)$ de uma árvore binária completa T. Começamos com a definição de altura de uma árvore binária completa.

DEFINIÇÃO 7 Definimos recursivamente a altura $h(T)$ de uma árvore binária completa T.

PASSO BASE: A altura da árvore binária completa T com apenas uma raiz r é $h(T) = 0$.

PASSO RECURSIVO: Se T_1 e T_2 forem árvores binárias completas, então a árvore binária completa $T = T_1 \cdot T_2$ tem altura $h(T) = 1 + \max(h(T_1), h(T_2))$.

Se considerarmos $n(T)$ como o número de vértices em uma árvore binária completa, observamos que $n(T)$ satisfaz à seguinte fórmula recursiva:

PASSO BASE: O número de vértices $n(T)$ da árvore binária completa T que tem apenas uma raiz r é $n(T) = 1$.

PASSO RECURSIVO: Se T_1 e T_2 forem árvores binárias completas, então o número de vértices da árvore binária completa $T = T_1 \cdot T_2$ é $n(T) = 1 + n(T_1) + n(T_2)$.

Mostraremos agora como a indução estrutural pode ser usada para demonstrar um resultado de árvores binárias completas.

TEOREMA 2 Se T for uma árvore binária completa T, então $n(T) \leq 2^{h(T)+1} - 1$.

Demonstração: Demonstramos essa inequação usando a indução estrutural.

PASSO BASE: Para a árvore binária completa que tem apenas a raiz r, o resultado é verdadeiro porque $n(T) = 1$ e $h(T) = 0$, assim, $n(T) = 1 \leq 2^{0+1} - 1 = 1$.

PASSO DE INDUÇÃO: Para a hipótese indutiva, assumimos que $n(T_1) \le 2^{h(T_1)+1} - 1$ e $n(T_2) \le 2^{h(T_2)+1} - 1$ sempre que T_1 e T_2 forem árvores binárias completas. Pelas fórmulas recursivas para $n(T)$ e $h(T)$, temos que $n(T) = 1 + n(T_1) + n(T_2)$ e $h(T) = 1 + \max(h(T_1), h(T_2))$.

Verificamos que

$$\begin{aligned}
n(T) &= 1 + n(T_1) + n(T_2) && \text{pela fórmula recursiva para } n(T)\\
&\le 1 + (2^{h(T_1)+1} - 1) + (2^{h(T_2)+1} - 1) && \text{pela hipótese indutiva}\\
&\le 2 \cdot \max(2^{h(T_1)+1}, 2^{h(T_2)+1}) - 1 && \text{porque a soma de dois termos é no máximo 2 vezes o maior}\\
&= 2 \cdot 2^{\max(h(T_1), h(T_2))+1} - 1 && \text{porque } \max(2^x, 2^y) = 2^{\max(x,y)}\\
&= 2 \cdot 2^{h(T)} - 1 && \text{pela definição recursiva de } h(T)\\
&= 2^{h(T)+1} - 1.
\end{aligned}$$

Isso completa o passo de indução. ◁

Indução Generalizada

Podemos estender a indução matemática para demonstrar os resultados de outros conjuntos que têm a propriedade da boa ordenação além do conjunto dos números inteiros. Discutiremos este conceito em detalhes na Seção 8.6, mas daremos alguns exemplos aqui para ilustrar a utilidade dessas aproximações.

Como exemplo, note que podemos definir uma ordem de $\mathbf{N} \times \mathbf{N}$, os pares ordenados dos números inteiros não negativos, especificando que (x_1, y_1) é menor que ou igual a (x_2, y_2) se $x_1 < x_2$ ou $x_1 = x_2$ e $y_1 < y_2$; isto é chamado de **ordem lexicográfica** (ordenação alfabética). O conjunto $\mathbf{N} \times \mathbf{N}$ com essa ordenação tem a propriedade de que todo subconjunto $\mathbf{N} \times \mathbf{N}$ tem pelo menos um elemento (veja o Exercício Complementar 53 da Seção 8.6). Isso implica que podemos definir recursivamente os termos $a_{m,n}$, com $m \in \mathbf{N}$ e $n \in \mathbf{N}$, e demonstrar os resultados deles usando uma variante da indução matemática, como ilustrado no Exemplo 15.

EXEMPLO 15 Suponha que $a_{m,n}$ seja definido recursivamente por $(m, n) \in \mathbf{N} \times \mathbf{N}$ por $a_{0,0} = 0$ e

$$a_{m,n} = \begin{cases} a_{m-1,n} + 1 & \text{se } n = 0 \text{ e } m > 0 \\ a_{m,n-1} + n & \text{se } n > 0. \end{cases}$$

Mostre que $a_{m,n} = m + n(n+1)/2$ para todo $(m, n) \in \mathbf{N} \times \mathbf{N}$, ou seja, para todos os pares de números inteiros não negativos.

Solução: Podemos demonstrar que $a_{m,n} = m + n(n+1)/2$ usando uma versão generalizada da indução matemática. O passo base requer que mostremos que esta fórmula é válida quando $(m, n) = (0, 0)$. O passo de indução requer que mostremos que se a fórmula for mantida para todos os pares menores que (m, n) em ordem lexicográfica de $\mathbf{N} \times \mathbf{N}$, então ela é também mantida para (m, n).

PASSO BASE: Considere $(m, n) = (0, 0)$. Então, pela passo base da definição recursiva de $a_{m,n}$, temos que $a_{0,0} = 0$. Além disso, quando $m = n = 0$, $m + n(n+1)/2 = 0 + (0 \cdot 1)/2 = 0$. Isso completa o passo base.

PASSO DE INDUÇÃO: Suponha que $a_{m',n'} = m' + n'(n'+1)/2$ sempre que (m', n') for menor que (m, n) na ordem lexicográfica de $\mathbf{N} \times \mathbf{N}$. Pela definição recursiva, se $n = 0$, então $a_{m,n} = a_{m-1,n} + 1$. Como $(m-1, n)$ é menor que (m, n), a hipótese indutiva nos diz que $a_{m-1,n} = m - 1 + n(n+1)/2$, assim, $a_{m,n} = m - 1 + n(n+1)/2 + 1 = m + n(n+1)/2$, resultando na equação desejada.

Agora suponha que $n > 0$, assim $a_{m,n} = a_{m,n-1} + n$. Como $(m, n-1)$ é menor que (m, n), a hipótese indutiva nos diz que $a_{m,n-1} = m + (n-1)n/2$, assim $a_{m,n} = m + (n-1)n/2 + n = m + (n^2 - n + 2n)/2 = m + n(n+1)/2$. Isso finaliza o passo de indução. ◀

Conforme mencionado, justificaremos esta técnica de demonstração na Seção 8.6.

Exercícios

1. Encontre $f(1)$, $f(2)$, $f(3)$ e $f(4)$ se $f(n)$ for definido recursivamente por $f(0) = 1$ e para $n = 0, 1, 2, \ldots$
- **a)** $f(n+1) = f(n) + 2$.
- **b)** $f(n+1) = 3f(n)$.
- **c)** $f(n+1) = 2^{f(n)}$.
- **d)** $f(n+1) = f(n)^2 + f(n) + 1$.

2. Encontre $f(1)$, $f(2)$, $f(3)$, $f(4)$ e $f(5)$ se $f(n)$ for definido recursivamente por $f(0) = 3$ e para $n = 0, 1, 2, \ldots$
- **a)** $f(n+1) = -2f(n)$.
- **b)** $f(n+1) = 3f(n) + 7$.
- **c)** $f(n+1) = f(n)^2 - 2f(n) - 2$.
- **d)** $f(n+1) = 3^{f(n)/3}$.

3. Encontre $f(2)$, $f(3)$, $f(4)$ e $f(5)$ se f for definido recursivamente por $f(0) = -1$, $f(1) = 2$ e para $n = 1, 2, \ldots$
- **a)** $f(n+1) = f(n) + 3f(n-1)$.
- **b)** $f(n+1) = f(n)^2 f(n-1)$.
- **c)** $f(n+1) = 3f(n)^2 - 4f(n-1)^2$.
- **d)** $f(n+1) = f(n-1)/f(n)$.

4. Encontre $f(2)$, $f(3)$, $f(4)$ e $f(5)$ se f for definido recursivamente por $f(0) = f(1) = 1$ e para $n = 1, 2, \ldots$
- **a)** $f(n+1) = f(n) - f(n-1)$.
- **b)** $f(n+1) = f(n)f(n-1)$.
- **c)** $f(n+1) = f(n)^2 + f(n-1)^3$.
- **d)** $f(n+1) = f(n)/f(n-1)$.

5. Determine se cada uma das definições propostas abaixo é uma definição recursiva válida de uma função f a partir do conjunto dos números inteiros não negativos para o conjunto dos números inteiros. Se f for bem definida, encontre uma fórmula para $f(n)$ quando n for um número inteiro não negativo e demonstre que sua fórmula é válida.
- **a)** $f(0) = 0$, $f(n) = 2f(n-2)$ para $n \geq 1$
- **b)** $f(0) = 1$, $f(n) = f(n-1) - 1$ para $n \geq 1$
- **c)** $f(0) = 2$, $f(1) = 3$, $f(n) = f(n-1) - 1$ para $n \geq 2$
- **d)** $f(0) = 1$, $f(1) = 2$, $f(n) = 2f(n-2)$ para $n \geq 2$
- **e)** $f(0) = 1$, $f(n) = 3f(n-1)$ se n for ímpar e $n \geq 1$ e $f(n) = 9f(n-2)$ se n for par e $n \geq 2$

6. Determine se cada uma das definições propostas a seguir é uma definição recursiva válida de uma função f a partir do conjunto dos números inteiros não negativos para o conjunto dos números inteiros. Se f for bem definida, encontre uma fórmula para $f(n)$ quando n for um número inteiro não negativo e demonstre que sua fórmula é válida.
- **a)** $f(0) = 1$, $f(n) = -f(n-1)$ para $n \geq 1$
- **b)** $f(0) = 1$, $f(1) = 0$, $f(2) = 2$, $f(n) = 2f(n-3)$ para $n \geq 3$
- **c)** $f(0) = 0$, $f(1) = 1$, $f(n) = 2f(n+1)$ para $n \geq 2$
- **d)** $f(0) = 0$, $f(1) = 1$, $f(n) = 2f(n-1)$ para $n \geq 1$
- **e)** $f(0) = 2$, $f(n) = f(n-1)$ se n for ímpar e $n \geq 1$ e $f(n) = 2f(n-2)$ se $n \geq 2$

7. Dê uma definição recursiva da seqüência $\{a_n\}$, $n = 1, 2, 3, \ldots$ se
- **a)** $a_n = 6n$.
- **b)** $a_n = 2n + 1$.
- **c)** $a_n = 10^n$.
- **d)** $a_n = 5$.

8. Dê uma definição recursiva da seqüência $\{a_n\}$, $n = 1, 2, 3, \ldots$ se
- **a)** $a_n = 4n - 2$.
- **b)** $a_n = 1 + (-1)^n$.
- **c)** $a_n = n(n+1)$.
- **d)** $a_n = n^2$.

9. Consulte F como uma função, tal que $F(n)$ é a soma dos primeiros n números inteiros positivos. Dê uma definição recursiva de $F(n)$.

10. Dê uma definição recursiva de $S_m(n)$, a soma do número inteiro m e do número inteiro não negativo n.

11. Dê uma definição recursiva de $P_m(n)$, o produto do número inteiro m pelo número inteiro não negativo n.

Nos exercícios 12 a 19, f_n é o n-ésimo número de Fibonacci.

12. Demonstre que $f_1^2 + f_2^2 + \cdots + f_n^2 = f_n f_{n+1}$ quando n é um número inteiro positivo.

13. Demonstre que $f_1 + f_3 + \cdots + f_{2n-1} = f_{2n}$ quando n é um número inteiro positivo.

***14.** Mostre que $f_{n+1}f_{n-1} - f_n^2 = (-1)^n$ quando n é um número inteiro positivo.

***15.** Mostre que $f_0 f_1 + f_1 f_2 + \cdots + f_{2n-1} f_{2n} = f_{2n}^2$ quando n é um número inteiro positivo.

***16.** Mostre que $f_0 - f_1 + f_2 - \cdots - f_{2n-1} + f_{2n} = f_{2n-1} - 1$ quando n é um número inteiro positivo.

17. Determine o número de divisões usadas pelo algoritmo de Euclides para encontrar o máximo divisor comum dos números de Fibonacci f_n e f_{n+1}, em que n é um número inteiro não negativo. Verifique sua resposta usando a indução matemática.

18. Considere

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

Mostre que

$$A^n = \begin{bmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{bmatrix}$$

quando n é um número inteiro positivo.

19. Considerando determinantes de ambos os lados da equação do Exercício 18, demonstre a identidade dada no Exercício 14. (Lembre-se de que o determinante de uma matriz $\begin{vmatrix} a & b \\ c & d \end{vmatrix}$ é $ad - bc$.)

***20.** Dê uma definição recursiva das funções max e min, para que $\max(a_1, a_2, \ldots, a_n)$ e $\min(a_1, a_2, \ldots, a_n)$ sejam o máximo e o mínimo de n números $a_1, a_2, \ldots, a_n$, respectivamente.

***21.** Considere $a_1, a_2, \ldots, a_n$ e $b_1, b_2, \ldots, b_n$ como números reais. Use as definições recursivas do Exercício 20 para demonstrar as equações abaixo.

a) $\max(-a_1, -a_2, \ldots, -a_n) = -\min(a_1, a_2, \ldots, a_n)$

b) $\max(a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n) \leq \max(a_1, a_2, \ldots, a_n) + \max(b_1, b_2, \ldots, b_n)$

c) $\min(a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n) \geq \min(a_1, a_2, \ldots, a_n) + \min(b_1, b_2, \ldots, b_n)$

22. Mostre que o conjunto S, definido por $1 \in S$ e $s + t \in S$ sempre que $s \in S$ e $t \in S$, é o conjunto dos números inteiros positivos.

23. Dê uma definição recursiva do conjunto dos números inteiros positivos que são múltiplos de 5.

24. Dê uma definição recursiva do

a) conjunto de números inteiros positivos e ímpares.

b) conjunto dos números inteiros positivos que são potências de 3.

c) conjunto de polinômios com coeficientes inteiros.

25. Dê uma definição recursiva do

a) conjunto de números inteiros pares.

b) conjunto de números inteiros positivos congruentes a 2 módulo 3.

c) conjunto de números inteiros positivos não divisíveis por 5.

26. Consulte S como o subconjunto do conjunto de pares ordenados de números inteiros definido recursivamente por

Passo base: $(0, 0) \in S$.

Passo recursivo: Se $(a, b) \in S$, então $(a + 2, b + 3) \in S$ e $(a + 3, b + 2) \in S$.

a) Liste os elementos de S produzidos pelas primeiras cinco aplicações do passo recursivo.

b) Use a indução completa no número de aplicações do passo recursivo da definição para mostrar que $5 \mid a + b$ quando $(a, b) \in S$.

c) Use a indução estrutural para mostrar que $5 \mid a + b$ quando $(a, b) \in S$.

27. Consulte S como o subconjunto do conjunto de pares ordenados de números inteiros definido recursivamente por

Passo base: $(0, 0) \in S$.

Passo recursivo: Se $(a, b) \in S$, então $(a, b + 1) \in S$, $(a + 1, b + 1) \in S$, e $(a + 2, b + 1) \in S$.

a) Liste os elementos de S produzidos pelas primeiras quatro aplicações do passo recursivo.

b) Use a indução completa sobre o número de aplicações do passo recursivo da definição para mostrar que $a \leq 2b$ sempre que $(a, b) \in S$.

c) Use a indução estrutural para mostrar que $a \leq 2b$ sempre que $(a, b) \in S$.

28. Dê uma definição recursiva para cada um dos conjuntos de pares ordenados de números inteiros positivos abaixo. [*Dica:* Organize os pontos do conjunto no plano e procure por linhas que contenham pontos no conjunto.]

a) $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+ \text{ e } a + b \text{ é impar}\}$

b) $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+ \text{ e } a \mid b\}$

c) $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+ \text{ e } 3 \mid a + b\}$

29. Dê uma definição recursiva para cada um dos conjuntos de pares ordenados de números inteiros positivos abaixo. Use a indução estrutural para demonstrar que a definição recursiva que você encontrou está correta. [*Dica:* Para encontrar uma definição recursiva, organize os pontos do conjunto no plano e procure por padrões.]

a) $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+ \text{ e } a + b \text{ é par}\}$

b) $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+ \text{ e } a \text{ ou } b \text{ é ímpar}\}$

c) $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+, a + b \text{ é impar e } 3 \mid b\}$

30. Demonstre que em uma cadeia de bits, a seqüência 01 aparece no máximo uma vez mais que a seqüência 10.

31. Defina as fórmulas bem formadas de conjuntos, variáveis que representam os conjuntos e os operadores de $\{\overline{\ }, \cup, \cap, -\}$.

32. **a)** Dê uma definição recursiva da função $uns(s)$, que contém o número de uns em uma cadeia de bits s.

b) Use a indução estrutural para demonstrar que $uns(st) = uns(s) + uns(t)$.

33. **a)** Dê uma definição recursiva da função $m(s)$, que é igual ao menor dígito em uma cadeia não vazia de dígitos decimais.

b) Use a indução estrutural para demonstrar que $m(st) = \min(m(s), m(t))$.

O **reverso** de uma cadeia é a cadeia que contém os símbolos daquela cadeia em ordem reversa. O reverso de uma cadeia w é indicado por w^R.

34. Encontre o reverso das cadeias de bits a seguir.

a) 0101 **b)** 1 1011 **c)** 1000 1001 0111

35. Dê uma definição recursiva do reverso de uma cadeia. [*Dica:* Primeiro, defina o reverso da cadeia vazia. Então, escreva uma cadeia w de extensão $n + 1$ como xy, em que x é uma cadeia de extensão n, e expresse o reverso de w em termos de x^R e y.]

***36.** Use a indução estrutural para demonstrar que $(w_1 w_2)^R = w_2^R w_1^R$.

37. Dê uma definição recursiva de w^i, em que w é uma cadeia e i é um número inteiro não negativo. (Aqui, w^i representa a concatenação de i cópias da cadeia w.)

***38.** Dê uma definição recursiva do conjunto de cadeias de bits que são palíndromos.

39. Quando uma cadeia pertence ao conjunto A de cadeias de bits definidas recursivamente por

$\lambda \in A$

$0x1 \in A$ se $x \in A$,

em que λ é a cadeia vazia?

***40.** Defina recursivamente o conjunto de cadeias de bits que têm mais zeros que uns.

41. Use o Exercício 37 e a indução matemática para mostrar que $l(w^i) = i \cdot l(w)$, em que w é uma cadeia e i é um número inteiro não negativo.

*42. Mostre que $(w^R)^i = (w^i)^R$ sempre que w for uma cadeia e i for um número inteiro não negativo; ou seja, mostre que a i-ésima potência da reversa de uma cadeia é a reversa da i-ésima potência da cadeia.

43. Use a indução estrutural para mostrar que $n(T) \geq 2h(T) + 1$, em que T é uma árvore binária completa, $n(T)$ é igual ao número de vértices de T e $h(T)$ é a altura de T.

O conjunto de folhas e o conjunto de vértices internos de uma árvore binária completa podem ser definidos recursivamente.

Passo base: A raiz r é uma folha da árvore binária completa com um vértice r. Esta árvore não tem vértices internos.

Passo recursivo: O conjunto de folhas da árvore $T = T_1 \cdot T_2$ é a união do conjunto de folhas de T_1 e o conjunto de folhas de T_2. Os vértices internos de T são a raiz r de T e a união do conjunto de vértices internos de T_1 e o conjunto de vértices internos de T_2.

44. Use a indução estrutural para mostrar que $l(T)$, o número de folhas de uma árvore binária completa T, é 1 mais $i(T)$, o número de vértices internos de T.

45. Use a indução generalizada, como foi feito no Exemplo 15, para mostrar que se $a_{m,n}$ for definido recursivamente por $a_{0,0} = 0$ e

$$a_{m,n} = \begin{cases} a_{m-1,n} + 1 & \text{se } n = 0 \text{ e } m > 0 \\ a_{m,n-1} + 1 & \text{se } n > 0, \end{cases}$$

então $a_{m,n} = m + n$ para todo $(m, n) \in \mathbf{N} \times \mathbf{N}$.

46. Use a indução generalizada, como foi feito no Exemplo 15, para mostrar que se $a_{m,n}$ for definido recursivamente por $a_{1,1} = 5$ e

$$a_{m,n} = \begin{cases} a_{m-1,n} + 2 & \text{se } n = 1 \text{ e } m > 1 \\ a_{m,n-1} + 2 & \text{se } n > 1, \end{cases}$$

então $a_{m,n} = 2(m + n) + 1$ para todo $(m, n) \in \mathbf{Z}^+ \times \mathbf{Z}^+$.

*47. Uma **partição** de um número inteiro positivo n é uma forma de escrever n como a soma de números inteiros positivos, em que a ordem dos termos na soma não importa. Por exemplo, $7 = 3 + 2 + 1 + 1$ é uma partição de 7. Considere P_m como o número de partições diferentes de m, e considere $P_{m,n}$ como os números de maneiras diferentes de expressar m como a soma dos números inteiros positivos não excedentes a n.

a) Mostre que $P_{m,m} = P_m$.

b) Mostre que a definição recursiva a seguir para $P_{m,n}$ está correta:

$$P_{m,n} = \begin{cases} 1 & \text{se } m = 1 \\ 1 & \text{se } n = 1 \\ P_{m,m} & \text{se } m < n \\ 1 + P_{m,m-1} & \text{se } m = n > 1 \\ P_{m,n-1} + P_{m-n,n} & \text{se } m > n > 1. \end{cases}$$

c) Encontre o número de partições de 5 e de 6 usando essa definição recursiva.

Considere a definição indutiva de uma versão da **função de Ackermann.** Essa função foi assim chamada em homenagem a Wilhelm Ackermann, um matemático alemão que foi aluno do grande matemático David Hilbert. A função de Ackermann tem um papel importante na teoria das funções recursivas e no estudo da complexidade de determinados algoritmos que envolvem a união de conjuntos. (Há muitas variações dessa função. Todas são chamadas de função de Ackermann e têm propriedades semelhantes, mesmo que seus valores nem sempre coincidam.)

$$A(m,n) = \begin{cases} 2n & \text{se } m = 0 \\ 0 & \text{se } m \geq 1 \text{ e } n = 0 \\ 2 & \text{se } m \geq 1 \text{ e } n = 1 \\ A(m-1, A(m,n-1)) & \\ & \text{se } m \geq 1 \text{ e } n \geq 2 \end{cases}$$

Os exercícios 48 a 55 envolvem essa versão da função de Ackermann.

48. Encontre os valores abaixo para a função de Ackermann.
a) $A(1, 0)$ **b)** $A(0, 1)$
c) $A(1, 1)$ **d)** $A(2, 2)$

49. Mostre que $A(m, 2) = 4$ sempre que $m \geq 1$.

50. Mostre que $A(1, n) = 2^n$ sempre que $n \geq 1$.

51. Encontre os valores abaixo para a função de Ackermann.
a) $A(2, 3)$ ***b)** $A(3, 3)$

*52. Encontre $A(3, 4)$.

**53. Demonstre que $A(m, n + 1) > A(m, n)$ sempre que m e n forem números inteiros não negativos.

*54. Demonstre que $A(m + 1, n) \geq A(m, n)$ sempre que m e n forem números inteiros não negativos.

55. Demonstre que $A(i, j) \geq j$ sempre que i e j forem números inteiros não negativos.

☞56. Use a indução matemática para demonstrar que uma função F definida especificando-se $F(0)$ e uma regra para obter $F(n + 1)$ de $F(n)$ é bem definida.

☞57. Use a indução completa para demonstrar que uma função F definida especificando-se $F(0)$ e uma regra para obter $F(n + 1)$ a partir dos valores de $F(k)$ para $k = 0, 1, 2, \ldots, n$ é bem definida.

58. Mostre que cada uma das definições recursivas propostas abaixo de uma função no conjunto dos números inteiros positivos não produz uma função bem definida.
a) $F(n) = 1 + F(\lfloor n/2 \rfloor)$ para $n \geq 1$ e $F(1) = 1$.
b) $F(n) = 1 + F(n - 3)$ para $n \geq 2$, $F(1) = 2$ e $F(2) = 3$.
c) $F(n) = 1 + F(n/2)$ para $n \geq 2$, $F(1) = 1$ e $F(2) = 2$.
d) $F(n) = 1 + F(n/2)$ se n for par e $n \geq 2$, $F(n) = 1 - F(n - 1)$ se n for ímpar e $F(1) = 1$.
e) $F(n) = 1 + F(n/2)$ se n for par e $n \geq 2$, $F(n) = F(3n - 1)$ se n for ímpar e $n \geq 3$ e $F(1) = 1$.

59. Mostre que cada uma das definições recursivas propostas abaixo de uma função no conjunto dos números inteiros positivos não produz uma função bem definida.
a) $F(n) = 1 + F(\lfloor (n + 1)/2 \rfloor)$ para $n \geq 1$ e $F(1) = 1$.
b) $F(n) = 1 + F(n - 2)$ para $n \geq 2$ e $F(1) = 0$.
c) $F(n) = 1 + F(n/3)$ para $n \geq 3$, $F(1) = 1$, $F(2) = 2$ e $F(3) = 3$.
d) $F(n) = 1 + F(n/2)$ se n for par e $n \geq 2$, $F(n) = 1 + F(n - 2)$ se n for ímpar e $F(1) = 1$.
e) $F(n) = 1 + F(F(n - 1))$ se $n \geq 2$ e $F(1) = 2$.

Os exercícios 60 a 62 lidam com iterações da função logarítmica. Considere $\log n$ como o logaritmo de n na base 2. A função $\log^{(k)} n$ é definida recursivamente por

$$\log^{(k)} n = \begin{cases} n & \text{se } k = 0 \\ \log(\log^{(k-1)} n) & \text{se } \log^{(k-1)} n \text{ é definido e positivo} \\ \text{indefinido} & \text{caso contrário.} \end{cases}$$

O **logaritmo iterado** é a função $\log^* n$ cujo valor em n é o menor número inteiro não negativo k, tal que $\log^{(k)} n \leq 1$.

60. Encontre cada um dos valores abaixo:
a) $\log^{(2)} 16$ **b)** $\log^{(3)} 256$
c) $\log^{(3)} 2^{65536}$ **d)** $\log^{(4)} 2^{2^{65536}}$

61. Encontre os valores de $\log^* n$ para cada um dos valores abaixo de n:
a) 2 **b)** 4 **c)** 8 **d)** 16
e) 256 **f)** 65536 **g)** 2^{2048}

62. Encontre o maior número inteiro n, tal que $\log^* n = 5$. Determine o número de dígitos decimais nesse número.

Os exercícios 63 a 65 tratam de funções com valores iterados. Suponha que $f(n)$ seja uma função do conjunto de números reais — ou números reais positivos ou algum outro conjunto de números reais — para o conjunto de números reais, tal que $f(n)$ é monotonicamente crescente [ou seja, $f(n) < f(m)$ quando $n < m$ e $f(n) < n$ para todo n no domínio de f.] A função $f^{(k)}(n)$ é definida recursivamente por

$$f^{(k)}(n) = \begin{cases} n & \text{se } k = 0 \\ f(f^{(k-1)}(n)) & \text{se } k > 0. \end{cases}$$

Além disso, considere c como um número real positivo. A **função iterada** f_c^* é o número de repetições de f necessário para reduzir seu argumento a c ou menos, assim $f_c^*(n)$ é o menor número inteiro não negativo k, tal que $f^k(n) \leq c$.

63. Considere $f(n) = n - a$, em que a é um número inteiro positivo. Encontre uma fórmula para $f^{(k)}(n)$. Qual o valor de $f_0^*(n)$ quando n for um número inteiro positivo?

64. Considere $f(n) = n/2$. Encontre uma fórmula para $f^{(k)}(n)$. Qual o valor de $f_1^*(n)$ quando n for um número inteiro positivo?

65. Considere $f(n) = \sqrt{n}$. Encontre uma formula para $f^{(k)}(n)$. Qual o valor de $f_2^*(n)$ quando n for um número inteiro positivo?

4.4 Algoritmos Recursivos

Introdução

Às vezes podemos reduzir a solução de um problema com um determinado conjunto de valores iniciais para a solução do mesmo problema com valores iniciais menores. Por exemplo, o problema para encontrar o máximo divisor comum de dois números inteiros positivos a e b, em que $b > a$, pode ser reduzido a encontrar o máximo divisor comum de um par de números inteiros menores, ou seja, $b \bmod a$ e a, pois $\text{mdc}(b \bmod a, a) = \text{mdc}(a, b)$. Quando essa redução pode ser feita, a solução do problema original pode ser encontrada com uma seqüência de reduções, até que o problema seja reduzido ao caso inicial para qualquer solução conhecida. Por exemplo, para encontrar o máximo divisor comum, a redução continua até que o menor dos dois números seja zero, porque $\text{mdc}(a, 0) = a$ quando $a > 0$.

Veremos os algoritmos que reduzem sucessivamente um problema ao mesmo problema com valores iniciais menores que são usados para resolver uma grande variedade de problemas.

DEFINIÇÃO 1 Um algoritmo é chamado de *recursivo* se resolver um problema reduzindo-o a um mesmo problema com valores iniciais menores.

Links Descreveremos vários algoritmos recursivos diferentes nesta seção.

EXEMPLO 1 Dê um algoritmo recursivo para computar $n!$, em que n é um número inteiro não negativo.

Exemplos Extras *Solução:* Podemos construir um algoritmo recursivo que encontre $n!$, em que n é um número inteiro não negativo, com base na definição recursiva de $n!$, que determina que $n! = n \cdot (n-1)!$ quando n for um número inteiro positivo e que $0! = 1$. Para encontrar $n!$ para determinado número inteiro, usamos o passo recursivo n vezes, cada vez substituindo um valor da função

fatorial pelo valor da função fatorial do próximo número inteiro menor. Nesta última etapa, inserimos o valor de 0!. O algoritmo recursivo que obtemos é mostrado como Algoritmo 1.

Para ajudar a entender como esse algoritmo funciona, traçamos estas etapas usadas pelo algoritmo para computar 4!. Primeiro, usamos o passo recursivo para escrever $4! = 4 \cdot 3!$. Então, usamos o mesmo passo recursivo repetidamente para escrever $3! = 3 \cdot 2!$, $2! = 2 \cdot 1!$, e $1! = 1 \cdot 0!$. Inserindo o valor de $0! = 1$ e trabalhando com os passos, vemos que $1! = 1 \cdot 1 = 1$, $2! = 2 \cdot 1! = 2$, $3! = 3 \cdot 2! = 3 \cdot 2 = 6$ e $4! = 4 \cdot 3! = 4 \cdot 6 = 24$. ◀

ALGORITMO 1 Um Algoritmo Recursivo para Computar $n!$.

procedure *fatorial*(n: número inteiro não negativo)
if $n = 0$ **then** *fatorial*(n) := 1
else *fatorial*(n) := $n \cdot$ *fatorial*($n - 1$)

O Exemplo 2 mostra como um algoritmo recursivo pode ser construído para avaliar uma função a partir de sua definição recursiva.

EXEMPLO 2 Dê um algoritmo recursivo para computar a^n, em que a é um número real diferente de zero e n é um número inteiro não negativo.

Solução: Podemos basear um algoritmo recursivo em uma definição recursiva de a^n. Essa definição afirma que $a^{n+1} = a \cdot a^n$ para $n > 0$ e a condição inicial $a^0 = 1$. Para encontrar a^n, usamos sucessivamente o passo recursivo para reduzir o expoente até que ele se torne zero. Apresentamos esse procedimento no Algoritmo 2. ◀

ALGORITMO 2 Um Algoritmo Recursivo para Computar a^n.

procedure *potencia*(a: número real diferente de zero, n: número inteiro não negativo)
if $n = 0$ **then** *potencia*(a, n) := 1
else *potencia* (a, n) := $a \cdot$ *potencia* (a, $n - 1$)

EXEMPLO 3 Construa um algoritmo recursivo para computar b^n **mod** m, em que b, n e m são números inteiros com $m \geq 2$, $n \geq 0$ e $1 \leq b < m$.

Solução: Podemos basear um algoritmo recursivo no fato de que

$$b^n \bmod m = (b \cdot (b^{n-1} \bmod m)) \bmod m,$$

que é mantido pelo Corolário 2 da Seção 3.4 e com condição inicial $b^0 \bmod m = 1$. Apresentamos essa questão no Exercício 12, ao final desta seção.

Entretanto, podemos construir algoritmos recursivos muito mais eficientes com base na observação de que

$$b^n \bmod m = (b^{n/2} \bmod m)^2 \bmod m$$

quando n é par e

$$b^n \bmod m = ((b^{\lfloor n/2 \rfloor} \bmod m)^2 \bmod m \cdot b \bmod m) \bmod m$$

quando n é ímpar, como descrito em pseudocódigo no Algoritmo 3.

Podemos rascunhar o Algoritmo 3 com a entrada $b = 2$, $n = 5$ e $m = 3$ para ilustrar como ele funciona. Primeiro, como $n = 5$ é ímpar, usamos a cláusula "else" para estabelecer que *mpotência* $(2, 5, 3) = (mpotência(2, 2, 3)^2$ **mod** $3 \cdot 2$ **mod** $3)$ **mod** 3. Depois, usamos a cláusula "else if" para ver que $mpotência(2, 2, 3) = mpotência(2, 1, 3)^2$ **mod** 3. Usando novamente "else", vemos que $mpotência(2, 1, 3) = (mpotência(2, 0, 3)^2$ **mod** $3 \cdot 2$ **mod** $3)$ **mod** 3. Por fim, usamos a sentença "if ", para ver que $mpotência(2, 0, 3) = 1$. Assim, temos que $mpotência(2, 1, 3) = (1^2$ **mod** $3 \cdot 2$ **mod** $3)$ **mod** $3 = 2$, então $mpotência(2, 2, 3) = 2^2$ **mod** $3 = 1$ e, finalmente, $mpotência(2, 5, 3) = (1^2$ **mod** $3 \cdot 2$ **mod** $3)$ **mod** $3 = 2$. ◀

ALGORITMO 3 Potenciação Modular Recursiva.

procedure *mpotencia*(b, n, m: números inteiros com $m \geq 2$, $n \geq 0$)
if $n = 0$ **then**
    $mpotencia(b, n, m) = 1$
else if n é par **then**
    $mpotencia(b, n, m) = mpotencia(b, n/2, m)^2$ **mod** m
else
    $mpotencia(b, n, m) = (mpotencia(b, \lfloor n/2 \rfloor, m)^2$ **mod** $m \cdot b$ **mod** $m)$ **mod** m
{$mpotencia(b, n, m) = b^n$ **mod** m}

Agora, apresentaremos um algoritmo recursivo para encontrar o máximo divisor comum.

EXEMPLO 4 Dê um algoritmo recursivo para computar o máximo divisor comum de dois números inteiros não negativos a e b com $a < b$.

Solução: Podemos basear o algoritmo recursivo na redução $mdc(a, b) = mdc(b$ **mod** $a, a)$ e na condição que $mdc(0, b) = b$ quando $b > 0$. Isso é feito no procedimento do Algoritmo 4, que é uma versão recursiva do algoritmo de Euclides.

Ilustramos o funcionamento do Algoritmo 4 calculando-o quando as entradas são $a = 5$, $b = 8$. Com essas entradas, o algoritmo usa a sentença "else" para encontrar $mdc(5, 8) = mdc(8$ **mod** $5, 5) = mdc(3, 5)$. Ele usa esta sentença novamente para encontrar $mdc(3, 5) = mdc(5$ **mod** $3, 3) = mdc(2, 3)$, então para ter $mdc(2, 3) = mdc(3$ **mod** $2, 2) = mdc(1, 2)$, então para ter $mdc(1, 2) = mdc(2$ **mod** $1, 1) = mdc(0, 1)$. Por fim, para encontrar $mdc(0, 1)$ ele usa o primeiro passo com $a = 0$ para encontrar $mdc(0, 1) = 1$. Conseqüentemente, o algoritmo encontra $mdc(5, 8) = 1$. ◀

ALGORITMO 4 Um Algoritmo Recursivo para Computar mdc(a, b).

procedure *mdc*(a, b: números inteiros não negativos com $a < b$)
if $a = 0$ **then** $mdc(a, b) := b$
else $mdc(a, b) := mdc(b$ **mod** $a, a)$

Vamos fornecer agora versões recursivas dos algoritmos de busca que foram introduzidos na Seção 3.1.

EXEMPLO 5 Expresse o algoritmo de busca linear como um procedimento recursivo.

Solução: Para a *busca* por x na seqüência de busca $a_1, a_2, \ldots, a_n$, no i-ésimo passo do algoritmo, x e a_i são comparados. Se x for igual a a_i, então i é a localização de x. Por outro lado, a busca por x é reduzida a uma busca em uma seqüência com um elemento a menos, ou seja, a seqüência $a_{i+1}, \ldots, a_n$. Podemos agora fornecer um procedimento recursivo, que está apresentado em pseudocódigo no Algoritmo 5.

Considere a *busca* (i, j, x) como o procedimento que procura por x na seqüência $a_i, a_{i+1}, \ldots, a_j$. A entrada do algoritmo consiste no trio $(1, n, x)$. O procedimento termina em um passo se o primeiro termo da seqüência restante for x ou se houver apenas um termo na seqüência e este não for x. Se x não for o primeiro termo e houver termos adicionais, o mesmo procedimento é realizado, mas com uma seqüência de busca com um termo a menos, obtido apagando-se o primeiro termo da seqüência de busca. ◀

ALGORITMO 5 Um Algoritmo de Busca Linear Recursivo.

procedure *busca*(i, j, x: i, j, x números inteiros, $1 \le i \le n$, $1 \le j \le n$)
if $a_i = x$ **then**
    localizacao := i
else if $i = j$ **then**
    localizacao := 0
else
    busca($i + 1, j, x$)

EXEMPLO 6 Construa uma versão recursiva do algoritmo de busca binária.

Solução: Suponha que queiramos localizar x na seqüência $a_1, a_2, \ldots, a_n$, números inteiros em ordem crescente. Para realizar uma busca binária, começamos pela comparação de x com o termo do meio, $a_{\lfloor (n+1)/2 \rfloor}$. Nosso algoritmo terminará se x for igual a este termo. Por outro lado, reduzimos a busca a uma seqüência de busca menor, ou seja, a primeira metade da seqüência se x for menor que o termo do meio da seqüência original, ou a segunda metade se ele for maior. Reduzimos a solução do problema de busca para a solução do mesmo problema como uma seqüência aproximadamente igual à metade da original. Expressamos esta versão recursiva do algoritmo de busca binária como Algoritmo 6. ◀

ALGORITMO 6 Um Algoritmo de Busca Binária Recursivo.

procedure *busca binaria*(i, j, x: i, j, x números inteiros, $1 \le i \le n$, $1 \le j \le n$)
$m := \lfloor (i + j)/2 \rfloor$
if $x = a_m$ **then**
    localizacao := m
else if ($x < a_m$ e $i < m$) **then**
    busca binaria($x, i, m - 1$)
else if ($x > a_m$ e $j > m$) **then**
    busca binaria($x, m + 1, j$)
else *localizacao* := 0

Demonstrando que os Algoritmos Recursivos Estão Corretos

A indução matemática e sua variante, a indução completa, podem ser usadas para demonstrar que um algoritmo recursivo está correto, ou seja, que ele produz a saída desejada para todos os valores de entrada possíveis. Os exemplos 7 e 8 ilustram como a indução matemática ou a completa podem ser usadas para demonstrar que os algoritmos recursivos estão corretos. Primeiro mostraremos que o Algoritmo 2 está correto.

EXEMPLO 7 Demonstre que o Algoritmo 2, que computa potências de números reais, está correto.

Solução: Usamos a indução matemática no expoente n.

PASSO BASE: Se $n = 0$, o primeiro passo do algoritmo nos diz que *potência* $(a, 0) = 1$. Isso é correto porque $a^0 = 1$ para todo número real diferente de zero a. Isso completa o passo base.

PASSO DE INDUÇÃO: A hipótese indutiva é a proposição de que a *potência* $(a, k) = a^k$ para todo $a \neq 0$ para o número inteiro não negativo k, ou seja, a hipótese indutiva é a proposição de que o algoritmo computará corretamente a^k. Para completar o passo de indução, mostramos que se a hipótese indutiva for verdadeira, então o algoritmo vai computar corretamente a^{k+1}. Como $k + 1$ é um número inteiro positivo, quando o algoritmo computar a^{k+1}, ele executará a *potência* $(a, k + 1) = a \cdot$ *potência* (a, k). Pela hipótese indutiva, temos *potência* $(a, k) = a^k$, assim, *potência* $(a, k + 1) = a \cdot$ *potência*$(a, k) = a \cdot a^k = a^{k+1}$. Isso completa o passo de indução.

Completamos os passos base e de indução; assim, podemos concluir que o Algoritmo 2 sempre computará a^n corretamente quando $a \neq 0$ e n for um número inteiro não negativo. ◀

Geralmente, precisamos usar a indução completa para demonstrar que os algoritmos recursivos estão corretos, em vez de utilizar apenas a indução matemática. O Exemplo 8 ilustra esse fato; ele mostra como a indução completa pode ser usada para demonstrar que o Algoritmo 3 está correto.

EXEMPLO 8 Demonstre que o Algoritmo 3, que faz a computação de potências modulares, está correto.

Exemplos Extras

Solução: Usamos a indução completa no expoente n.

PASSO BASE: Considere b como um número inteiro e m como um número inteiro com $m \geq 2$. Quando $n = 0$, o algoritmo executa *mpotência*(b, n, m) igual a 1. Isto está correto porque $b^0 \bmod m = 1$. O passo base está completo.

PASSO DE INDUÇÃO: Para a hipótese indutiva, assuma que *mpotência*$(b, j, m) = b^j \bmod m$ para todos os números inteiros $0 \leq j < k$ sempre que b for um número inteiro positivo e m for um número inteiro com $m \geq 2$. Para completar o passo de indução, mostramos que se ele estiver correto, então *mpotência*$(b, k, m) = b^k \bmod m$. Como o algoritmo recursivo trata de valores pares e ímpares de k de forma diferente, separamos o passo de indução em dois casos.

Quando k é par, temos

$$mpotência(b, k, m) = mpotência(b, k/2, m)^2 \bmod m = (b^{k/2} \bmod m)^2 \bmod m = b^k \bmod m,$$

em que usamos a hipótese indutiva para substituir *mpotência*$(b, k/2, m)$ por $b^{k/2} \bmod m$.

Quando k é ímpar, temos

$$\begin{aligned} mpotência(b, k, m) &= ((mpotência(b, \lfloor k/2 \rfloor, m))^2 \bmod m \cdot b \bmod m) \bmod m \\ &= ((b^{\lfloor k/2 \rfloor} \bmod m)^2 \bmod m \cdot b \bmod m) \bmod m \\ &= b^{2\lfloor k/2 \rfloor + 1} \bmod m = b^k \bmod m, \end{aligned}$$

usando o Corolário 2 da Seção 3.4, pois $2\lfloor k/2 \rfloor + 1 = 2(k-1)/2 + 1 = k$ quando k é ímpar. Aqui nós usamos a hipótese indutiva para substituir *mpotência*$(b, \lfloor k/2 \rfloor, m)$ por $b^{\lfloor k/2 \rfloor} \bmod m$. Isso completa o passo de indução.

Completamos os passos base e o de indução. Assim, pela indução completa, sabemos que o Algoritmo 3 está correto. ◀

Recursão e Iteração

Uma definição recursiva expressa o valor de uma função de um número inteiro positivo em termos de valores da função em números inteiros menores. Isso significa que podemos construir um algoritmo recursivo a fim de avaliar uma função definida recursivamente para um número inteiro positivo. Em vez de reduzir a computação sucessivamente para calcular o valor da função em números inteiros menores, podemos começar com o valor da função em um ou mais inteiros, os casos de base, e aplicar a definição recursiva sucessivamente para encontrar os valores da função dos números inteiros maiores. Esse procedimento é chamado de **iteração.** Geralmente um método iterativo para a valoração de uma seqüência definida recursivamente requer menos computações que um procedimento que usa a recursão (a menos que máquinas especiais para esse propósito sejam usadas). Isso é ilustrado pelos procedimentos de iteração e recursão para encontrar o n-ésimo número de Fibonacci. O procedimento recursivo é apresentado primeiro.

ALGORITMO 7 Um Algoritmo Recursivo para Números de Fibonacci.

procedure *fibonacci*(n: número inteiro não negativo)
if $n = 0$ **then** *fibonacci*$(0) := 0$
else if $n = 1$ **then** *fibonacci*$(1) := 1$
else *fibonacci*$(n) :=$ *fibonacci*$(n - 1) +$ *fibonacci*$(n - 2)$

Quando usamos um procedimento recursivo para encontrar f_n, primeiro expressamos f_n como $f_{n-1} + f_{n-2}$. Então, substituímos esses dois números de Fibonacci pela soma de dois números de Fibonacci anteriores, e assim por diante. Quando f_1 ou f_0 aparecerem, a soma é substituída por seu valor.

Note que em cada estágio da recursão, até que f_1 ou f_0 seja obtido, os números de Fibonacci que serão avaliados dobram. Por exemplo, quando encontramos f_4 usando este algoritmo recursivo, devemos realizar a computação ilustrada no diagrama de árvore da Figura 1. Esta árvore consiste em uma raiz marcada por f_4 e galhos a partir da raiz até os vértices marcados com os dois números de Fibonacci f_3 e f_2 que aparecem na redução da computação de f_4. Cada subseqüência reduzida produz dois galhos na árvore. Os galhos terminam quando f_0 e f_1 são alcançados. O leitor pode verificar que este algoritmo necessita de $f_{n+1} - 1$ adições para encontrar f_n.

Agora considere a quantidade de computação necessária para encontrar f_n usando o método iterativo no Algoritmo 8.

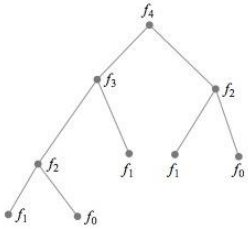


FIGURA 1 Cálculo de f_4 Recursivamente.

ALGORITMO 8 Um Algoritmo de Iteração para Computar Números de Fibonacci.

```
procedure iteracao de fibonacci(n: número inteiro não negativo)
if n = 0 then y := 0
else
begin
    x := 0
    y := 1
    for i := 1 to n − 1
    begin
        z := x + y
        x := y
        y := z
    end
end
{y é o n-ésimo número de Fibonacci}
```

Este procedimento inicia x como $f_0 = 0$ e y como $f_1 = 1$. Quando o laço é transversal, a soma de x e y é determinada pela variável auxiliar z. Então, x é substituído pelo valor de y e y é substituído pelo valor da variável auxiliar z. Então, depois de passar pelo laço a primeira vez, temos que x é igual a f_1 e y é igual a $f_0 + f_1 = f_2$. Além disso, depois de passar pelo laço $n - 1$ vezes, x é igual a f_{n-1} e y é igual a f_n (o leitor pode verificar esta proposição). Apenas $n - 1$ adições foram usadas para encontrar f_n com este método iterativo quando $n > 1$. Conseqüentemente, este algoritmo necessita de menos computações que o algoritmo recursivo.

Mostramos que um algoritmo recursivo pode necessitar de mais computações que um algoritmo de iteração, quando uma função definida recursivamente é avaliada. Às vezes, é preferível usar um procedimento recursivo, mesmo que ele seja menos eficiente que o de iteração. Em particular, isto é verdadeiro quando o método recursivo é facilmente implementado e o de iteração não. (Além disso, máquinas desenvolvidas recursivamente podem estar disponíveis, o que elimina a vantagem de usar a iteração.)

A Merge Sort

Descreveremos agora um algoritmo de ordenação recursivo chamado de algoritmo **merge sort**. Vamos demonstrar como ele funciona com um exemplo antes de descrevê-lo.

EXEMPLO 9 Ordene a lista 8, 2, 4, 6, 9, 7, 10, 1, 5, 3 usando a merge sort.

Solução: Uma merge sort começa separando a lista em elementos individuais por sucessivas separações de dois em dois. A progressão das sublistas deste exemplo está representada com a árvore binária de altura 4 mostrada na primeira parte da Figura 2.

O ordenamento é feito pela união sucessiva dos pares das listas. No primeiro estágio, os pares dos elementos individuais são unidos em listas de extensão dois em ordem crescente. Então, sucessivas uniões de pares de listas são realizadas até que toda a lista seja colocada em ordem crescente. A sucessão de listas unidas em ordem crescente é representada pela árvore binária de altura 4 mostrada na segunda parte da Figura 2 (note que essa árvore está disposta "de ponta-cabeça"). ◀

No geral, uma merge sort é realizada pela divisão das listas repetidamente em duas sublistas de igual extensão (ou uma sublista tem um elemento a mais que a outra) até cada sub-lista ter um elemento. Essa sucessão de sublistas pode ser representada por uma árvore binária. O

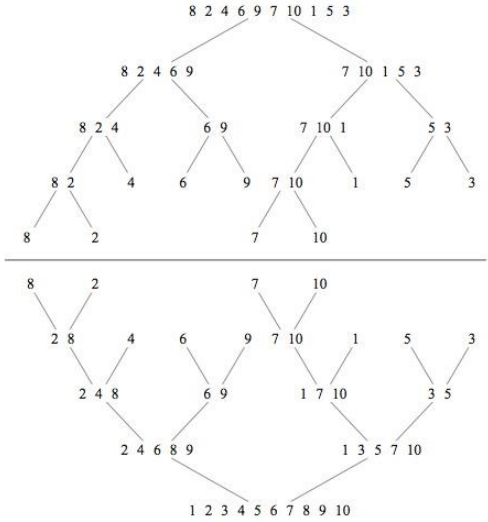


FIGURA 2 **A Merge Sort de 8, 2, 4, 6, 9, 7, 10, 1, 5, 3.**

Demo

procedimento continua unindo sucessivamente os pares das listas, em que ambas as listas estão em ordem crescente, em uma lista maior com os elementos em ordem crescente, até que a lista original seja colocada em ordem crescente. A sucessão de listas unidas pode ser representada por uma árvore binária.

Podemos também descrever a merge sort recursivamente. Para fazer uma merge sort, separamos uma lista em duas sublistas de igual, ou aproximadamente igual, tamanho, ordenando cada uma delas por meio do algoritmo de merge sort e, então, unindo as duas listas. A versão recursiva da merge sort é dada pelo Algoritmo 9. Este algoritmo usa a sub-rotina *merge*, que é descrita no Algoritmo 10.

ALGORITMO 9 **Uma Merge Sort Recursiva.**

procedure *mergesort*($L = a_1, \ldots, a_n$)
if $n > 1$ **then**
  $m := \lfloor n/2 \rfloor$
  $L_1 := a_1, a_2, \ldots, a_m$
  $L_2 := a_{m+1}, a_{m+2}, \ldots, a_n$
  $L :=$ *merge*(*mergesort*(L_1), *mergesort*(L_2))
{L é agora ordenado em elementos em ordem não decrescente}

TABELA 1 Unindo Duas Listas Ordenadas 2, 3, 5, 6 e 1, 4.

Primeira Lista	*Segunda Lista*	*Lista Unida*	*Comparação*
2 3 5 6	1 4		$1 < 2$
2 3 5 6	4	1	$2 < 4$
3 5 6	4	1 2	$3 < 4$
5 6	4	1 2 3	$4 < 5$
5 6		1 2 3 4	
		1 2 3 4 5 6	

Um algoritmo eficiente para unir duas listas ordenadas em uma lista ordenada maior é necessário para implementar a merge sort. Descreveremos agora este procedimento.

EXEMPLO 10 Una as duas listas: 2, 3, 5, 6 e 1, 4.

Solução: A Tabela 1 ilustra os passos a ser seguidos. Primeiro, compare os menores elementos das duas listas, 2 e 1, respectivamente. Como 1 é o menor, coloque-o no começo da lista unida e remova-o da segunda lista. Neste passo, a primeira lista é 2, 3, 5, 6, a segunda lista, 4 e a lista combinada (unida) é 1.

Depois, compare 2 e 4, os menores elementos das duas listas. Como 2 é o menor, adicione-o à lista combinada e remova-o da primeira lista. Neste estágio, a primeira lista é 3, 5, 6, a segunda lista é 4 e a lista combinada é 1, 2.

Continue comparando os menores elementos de suas respectivas listas, 3 e 4. Como 3 é o menor desses dois elementos, adicione-o à lista combinada e remova-o da primeira lista. Nesse passo, a primeira lista é 5, 6 e a segunda lista é 4. A lista combinada é 1, 2, 3.

Então, compare 5 e 4, os menores elementos das duas listas. Como 4 é o menor elemento, adicione-o à lista combinada e remova-o da segunda lista. Neste estágio, a primeira lista é 5, 6, a segunda está vazia e a lista combinada é 1, 2, 3, 4.

Por fim, como a segunda lista está vazia, todos os elementos da primeira lista podem ser colocados no fim da lista combinada na ordem em que eles aparecem na primeira lista. Isto produz a lista ordenada 1, 2, 3, 4, 5, 6. ◀

Vamos agora considerar o problema geral da união de duas listas ordenadas L_1 e L_2 em uma lista ordenada L. Vamos descrever um algoritmo para resolver esse problema. Comece com uma lista vazia L. Compare os menores elementos das duas listas. Coloque o menor desses elementos no extremo esquerdo de L e remova-o da lista em que estava. Depois, se L_1 ou L_2 estiverem vazias, insira a lista (não vazia) em L, completando a união. Se nem L_1 nem L_2 estiverem vazias, repita esse processo. O Algoritmo 10 dá uma descrição em pseudocódigo desse procedimento.

Precisaremos estimar o número de comparações usadas para unir duas listas ordenadas na análise da merge sort (ordenação por fusão). Podemos obter facilmente essa estimativa a partir do Algoritmo 10. Cada vez que uma comparação entre um elemento de L_1 e um elemento de L_2 for feita, um elemento é adicionado à lista L. Entretanto, quando L_1 ou L_2 estiverem vazias, não serão necessárias mais comparações. Assim, o Algoritmo 10 é menos eficiente quando $m + n - 2$ comparações são realizadas, em que m e n são o número de elementos em L_1 e L_2, respectivamente, deixando um elemento em L_1 e um em L_2. A próxima comparação será a última necessária, pois uma dessas listas ficará vazia. Assim, o Algoritmo 10 usa não mais que $m + n - 1$ comparações. O Lema 1 resume essa estimativa.

ALGORITMO 10 Unindo Duas Listas.

procedure *merge*(L_1, L_2 : listas ordenadas)
L := lista vazia
while L_1 e L_2 não são vazias
begin
  remova o menor entre os primeiros elementos de L_1 e L_2
  e coloque-o à direita em L
  if remover esse elemento torna uma lista vazia **then** remova
  todos os elementos da outra lista e insira-os em L
end {L é a lista unida com elementos em ordem crescente}

LEMA 1 Duas listas ordenadas com m elementos e n elementos podem ser unidas em uma lista ordenada usando não mais que $m + n - 1$ comparações.

Às vezes, duas listas ordenadas de extensão m e n podem ser unidas usando menos que $m + n - 1$ comparações. Por exemplo, quando $m = 1$, um procedimento de busca binária pode ser aplicado para colocar um elemento da primeira lista na segunda. Isso necessita apenas $\lceil \log n \rceil$ comparações, que é muito menor que $m + n - 1 = n$, para $m = 1$. Por outro lado, para alguns valores de m e n, o Lema 1 oferece a melhor margem possível, ou seja, existem listas com m e n elementos que não podem ser unidas usando menos que $m + n - 1$ comparações. (Veja o Exercício 47 no final desta seção.)

Podemos agora analisar a complexidade da merge sort. Em vez de estudarmos o problema geral, assumimos que n, o número elementos na primeira lista, é uma potência de 2, digamos 2^m. Isso tornará a análise menos complicada, mas, quando este não for o caso, várias modificações que vão apontar a mesma estimativa podem ser aplicadas.

No primeiro estágio do processo de separação, a lista é dividida em duas sublistas de 2^{m-1} elementos cada, no nível 1 da árvore construída pela divisão. Esse processo continua dividindo as duas sublistas com 2^{m-1} elementos em quatro sublistas com 2^{m-2} elementos cada no nível 2, e assim por diante. No geral, há 2^{k-1} listas no nível $k-1$, cada uma com 2^{m-k+1} elementos. Essas listas no nível $k-1$ são divididas em 2^k listas, no nível k, cada uma com 2^{m-k} elementos. No final desse procedimento, temos 2^m listas, cada uma com um elemento no nível m.

Começamos unindo (combinando) os pares das 2^m listas de um elemento em 2^{m-1} listas, no nível $m-1$, cada uma com dois elementos. Para fazer isso, 2^{m-1} pares de listas com um elemento cada são unidas. A união de cada par necessita de exatamente uma comparação.

O procedimento continua, até que no nível k ($k = m, m-1, m-2, \ldots, 3, 2, 1$), 2^k listas, cada uma com 2^{m-k} elementos, são unidas em 2^{k-1} listas, cada uma com 2^{m-k+1} elementos, no nível $k-1$. Para fazer isso, um total de 2^{k-1} uniões das duas listas, cada uma com 2^{m-k} elementos, são necessárias. Contudo, pelo Lema 1, cada uma dessas uniões pode ser realizada usando no máximo $2^{m-k} + 2^{m-k} - 1 = 2^{m-k+1} - 1$ comparações. Assim, pode-se passar do nível k para o $k-1$ usando-se no máximo $2^{k-1}(2^{m-k+1} - 1)$ comparações.

Somando algumas dessas estimativas, verificamos que o número de comparações necessárias para a merge sort é no máximo

$$\sum_{k=1}^{m} 2^{k-1}(2^{m-k+1} - 1) = \sum_{k=1}^{m} 2^m - \sum_{k=1}^{m} 2^{k-1} = m2^m - (2^m - 1) = n\log n - n + 1,$$

pois $m = \log n$ e $n = 2m$. (Calculamos $\Sigma_{k=1}^{m} 2^m$ notando que ele é a soma de m termos idênticos, cada um igual a 2^m. Calculamos $\Sigma_{k=1}^{m} 2^{k-1}$ usando a fórmula para a soma dos termos de uma progressão geométrica do Teorema 1 da Seção 2.4.)

Esta análise mostra que a merge sort alcança a melhor estimativa O grande possível para o número de comparações necessárias pelos algoritmos de ordenação, como afirmado no teorema a seguir.

TEOREMA 1 O número de comparações necessárias para a merge sort ordenar uma lista com n elementos é $O(n \log n)$.

Descreveremos outro algoritmo eficiente, a quick sort (ou ordenação rápida), ao longo dos exercícios.

Exercícios

1. Trace o Algoritmo 1 quando é dado $n = 5$ como entrada, ou seja, mostre todos os passos usados pelo Algoritmo 1 para encontrar 5!, como foi feito no Exemplo 1 para encontrar 4!.
2. Trace o Algoritmo 1 quando é dado $n = 6$ como entrada, ou seja, mostre todos os passos usados pelo Algoritmo 1 para encontrar 6!, como foi feito no Exemplo 1 para encontrar 4!.
3. Trace o Algoritmo 3 quando é dado $m = 5$, $n = 11$ e $b = 3$ como entradas, ou seja, mostre todos os passos que o Algoritmo 3 usa para encontrar 3^{11} **mod** 5.
4. Trace o Algoritmo 3 quando é dado $m = 7$, $n = 10$ e $b = 2$ como entradas, ou seja, mostre todos os passos que o Algoritmo 3 usa para encontrar 2^{10} **mod** 7.
5. Trace o Algoritmo 4 para encontrar mdc(8, 13), ou seja, mostre todos os passos que o Algoritmo 4 usa para encontrar mdc(8, 13).
6. Trace o Algoritmo 4 para encontrar mdc(12, 17), ou seja, mostre todos os passos usados pelo Algoritmo 4 para encontrar mdc(12, 17).
7. Dê um algoritmo recursivo para a computação de nx sempre que n for um número inteiro positivo e x for um número inteiro, usando apenas a adição.
8. Dê um algoritmo recursivo para encontrar a soma dos primeiros n números inteiros positivos.
9. Dê um algoritmo recursivo para encontrar a soma dos primeiros n números inteiros positivos e ímpares.
10. Dê um algoritmo recursivo para encontrar o máximo de um conjunto finito de números inteiros, considerando o fato de que o máximo de n números inteiros é o maior entre o último inteiro da lista e o máximo dos primeiros $n - 1$ números inteiros da lista.
11. Dê um algoritmo recursivo para encontrar o mínimo de um conjunto finito de números inteiros, considerando o fato de que o mínimo de n números inteiros é o menor entre o último inteiro da lista e o mínimo dos primeiros $n - 1$ números inteiros da lista.
12. Construa um algoritmo recursivo para encontrar x^n **mod** m sempre que n, x e m forem números inteiros positivos com base no fato de que x^n **mod** $m = (x^{n-1}$ **mod** $m \cdot x$ **mod** $m)$ **mod** m.
13. Dê um algoritmo recursivo para encontrar $n!$ **mod** m sempre que n e m forem números inteiros positivos.
14. Dê um algoritmo recursivo para encontrar uma **moda** de uma lista de números inteiros. (Uma **moda** é um elemento na lista que aparece ao menos tanto quanto os outros elementos.)
15. Construa um algoritmo recursivo para a computação do máximo divisor comum de dois números inteiros não negativos a e b com $a < b$, considerando o fato de que mdc(a, b) = mdc$(a, b - a)$.
16. Demonstre que o algoritmo recursivo usado para encontrar a soma dos primeiros n números inteiros positivos que você desenvolveu no Exercício 8 está correto.
17. Descreva um algoritmo recursivo para multiplicar dois números inteiros não negativos x e y com base no fato de que $xy = 2(x \cdot (y/2))$ quando y é par e $xy = 2(x \cdot \lfloor y/2 \rfloor) + x$ quando y é ímpar, junto com a condição inicial $xy = 0$ quando $y = 0$.
18. Demonstre que o Algoritmo 1 para a computação de $n!$ quando n é um número inteiro não negativo está correto.
19. Demonstre que o Algoritmo 4 para a computação de mdc(a, b) quando a e b são números inteiros positivos com $a < b$ está correto.
20. Demonstre que o algoritmo que você desenvolveu no Exercício 11 está correto.
21. Demonstre que o algoritmo recursivo que você desenvolveu no Exercício 7 está correto.
22. Demonstre que o algoritmo recursivo que você desenvolveu no Exercício 10 está correto.
23. Desenvolva um algoritmo recursivo para a computação de n^2, em que n é um número inteiro não negativo considerando o fato de que $(n + 1)^2 = n^2 + 2n + 1$. Então, demonstre que esse algoritmo está correto.
24. Desenvolva um algoritmo recursivo para encontrar a^{2^n}, em que a é um número real e n é um número inteiro positivo. [*Dica:* Use a equação $a^{2^{n+1}} = (a^{2^n})^2$.]
25. Como o número de multiplicações usadas pelo algoritmo do Exercício 24 pode ser comparado ao número de multiplicações usadas pelo Algoritmo 2 para avaliar a^{2^n}?

*26. Use o algoritmo do Exercício 24, a fim de desenvolver um algoritmo para avaliar a^n quando n for um número inteiro não negativo. [*Dica:* Use a expansão binária de n.]

*27. Como o número de multiplicações usadas pelo algoritmo do Exercício 26 pode ser comparado ao número de multiplicações usadas pelo Algoritmo 2 para avaliar a^n?

28. Quantas adições são usadas pelos algoritmos recursivo e de iteração dados nos Algoritmos 7 e 8, respectivamente, para encontrar o número de Fibonacci f_7?

29. Desenvolva um algoritmo recursivo para encontrar o n-ésimo termo da seqüência definida por $a_0 = 1$, $a_1 = 2$ e $a_n = a_{n-1} \cdot a_{n-2}$, para $n = 2, 3, 4, \ldots$.

30. Desenvolva um algoritmo de iteração para encontrar o n-ésimo termo da seqüência definida no Exercício 29.

31. Para encontrar a seqüência do Exercício 29, é mais eficiente o algoritmo recursivo ou o de iteração?

32. Desenvolva um algoritmo recursivo para encontrar o n-ésimo termo da seqüência definida por $a_0 = 1$, $a_1 = 2$, $a_2 = 3$ e $a_n = a_{n-1} + a_{n-2} + a_{n-3}$, para $n = 3, 4, 5, \ldots$.

33. Desenvolva um algoritmo de iteração para encontrar o n-ésimo termo da seqüência definida no Exercício 32.

34. Para encontrar a seqüência do Exercício 32, é mais eficiente o algoritmo recursivo ou o de iteração?

35. Dê os algoritmos recursivos e de iteração para encontrar o n-ésimo termo da seqüência definida por $a_0 = 1$, $a_1 = 3$, $a_2 = 5$ e $a_n = a_{n-1} \cdot a^2_{n-2} \cdot a^3_{n-3}$. Qual é o mais eficiente?

36. Dê um algoritmo recursivo para encontrar o número de partições de um número inteiro positivo com base na definição recursiva apresentada no Exercício 47 da Seção 4.3.

37. Dê um algoritmo recursivo para encontrar a reversa de uma cadeia de bits. (Veja a definição de reversa de uma cadeia de bits no prêambulo do Exercício 34 da Seção 4.3.)

38. Dê um algoritmo recursivo para encontrar a cadeia w^i, a concatenação de i cópias de w, quando w é uma cadeia de bits.

39. Demonstre que o algoritmo recursivo usado para encontrar a reversa de uma cadeia de bits que você desenvolveu no Exercício 37 está correto.

40. Demonstre que o algoritmo recursivo usado para encontrar a concatenação de i cópias de uma cadeia de bits que você desenvolveu no Exercício 38 está correto.

***41.** Dê um algoritmo recursivo para ladrilhar um tabuleiro de damas $2^n \times 2^n$ com um quadrado faltando, usando triominós.

42. Dê um algoritmo recursivo para triangular um polígono simples com n lados, usando o Lema 1 da Seção 4.2.

43. Dê um algoritmo recursivo para a computação dos valores da função de Ackermann. [*Dica:* Veja o preâmbulo do Exercício 48 da Seção 4.3.]

44. Use uma merge sort para ordenar 4, 3, 2, 5, 1, 8, 7, 6. Mostre todos os passos usados pelo algoritmo.

45. Use uma merge sort para ordenar b, d, a, f, g, h, z, p, o, k. Mostre todos os passos usados pelo algoritmo.

46. Quantas comparações são necessárias para unir os pares de listas abaixo usando o Algoritmo 10?

a) 1, 3, 5, 7, 9; 2, 4, 6, 8, 10

b) 1, 2, 3, 4, 5; 6, 7, 8, 9, 10

c) 1, 5, 6, 7, 8; 2, 3, 4, 9, 10

47. Mostre que para todos os números inteiros positivos m e n há listas com m elementos e n elementos, respectivamente, tal como o Algoritmo 10 usa $m + n - 1$ comparações para unilas em uma lista ordenada.

***48.** Qual é o menor número de comparações necessárias para unir quaisquer duas listas em uma lista de ordem crescente quando o número de elementos das duas listas são

a) 1, 4? **b)** 2, 4? **c)** 3, 4? **d)** 4, 4?

***49.** Demonstre que o algoritmo de merge sort está correto.

A quick sort – ou **ordenação rápida** – é um algoritmo eficiente. Para ordenar $a_1, a_2, \ldots, a_n$, esse algoritmo começa considerando o primeiro elemento a_1 e formando duas sublistas, a primeira contém aqueles elementos que são menores que a_1, na ordem em que eles aparecem, e a segunda contém aqueles elementos maiores que a_1, na ordem em que eles aparecem. Então, a_1 é posto no final da primeira sublista. Esse procedimento é repetido recursivamente para cada sublista, até que todas as sublistas contenham um item. A lista ordenada de n itens é obtida juntando-se as sublistas na ordem em que os elementos aparecem.

50. Ordene 3, 5, 7, 8, 1, 9, 2, 4, 6 usando a quick sort.

51. Considere $a_1, a_2, \ldots, a_n$ como uma lista de n números reais distintos. Quantas comparações são necessárias para formar duas sublistas a partir dessa lista, a primeira com elementos que são menores que a_1 e a segunda com elementos maiores que a_1?

52. Descreva o algoritmo quick sort usando o pseudocódigo.

53. Qual é o maior número de comparações necessárias para ordenar uma lista de quatro elementos usando o algoritmo quick sort?

54. Qual é o menor número de comparações necessárias para ordenar uma lista de quatro elementos usando o algoritmo quick sort?

55. Determine a complexidade de pior caso do algoritmo quick sort em termos do número de comparações usadas.

4.5 Exatidão de Programas

Introdução

Suponha que criamos um algoritmo para resolver um problema e escrevemos um programa para implementá-lo. Como podemos ter certeza de que o programa sempre produzirá a resposta correta? Depois que todos os erros forem removidos para que a sintaxe esteja correta, podemos testar o programa com entradas de amostra. Ele não está correto, se for produzido um resultado incorreto para qualquer entrada de amostra. Entretanto, mesmo se o programa der a resposta correta para todas as entradas de amostra, ele pode nem sempre produzir a resposta correta (a menos que todas as entradas possíveis forem testadas). Precisamos de uma demonstração para mostrar que o programa *sempre* fornece a saída correta.

A verificação de programas, ou a demonstração da exatidão dos programas, usa as regras de inferência e as técnicas de demonstração descritas neste capítulo, incluindo a indução matemática.

Visto que um programa incorreto pode levar a resultados desastrosos, várias metodologias têm sido construídas para verificar programas. Esforços têm sido feitos para automatizar a verificação de programas para que ela possa ser realizada por meio de um computador. Entretanto, apenas um progresso limitado foi alcançado nesse sentido. De fato, alguns matemáticos e teóricos da ciência da computação argumentam que nunca será real a mecanização da demonstração para verificar a exatidão de programas complexos.

Alguns dos conceitos e métodos usados para demonstrar que os programas estão corretos serão introduzidos nesta seção. Muitos métodos diferentes foram desenvolvidos para isso. Discutiremos o método introduzido por Tony Hoare, mas muitos outros foram desenvolvidos. Além disso, não vamos desenvolver uma metodologia completa para a verificação de programas neste livro. O objetivo desta seção é apresentar uma breve introdução à área da verificação de programas, juntamente com as regras de lógica, as técnicas de demonstração e o conceito de algoritmo.

Verificação de Programas

Dizemos que um programa está **correto** se ele produzir uma saída correta para toda entrada possível. A demonstração de que um programa está correto é feita em duas partes. A primeira parte mostra que a resposta correta é obtida se o programa terminar. Essa parte da demonstração estabelece a **exatidão parcial** do programa. A segunda parte da demonstração mostra que o programa termina.

Para especificar o que significa um programa produzir a saída correta, duas proposições serão usadas. A primeira é a **asserção inicial**, que dá as propriedades que os valores de entrada devem ter. A segunda é a **asserção final**, que dá as propriedades que os valores de saída do programa devem ter, se o programa fizer o que é esperado. As asserções inicial e final apropriadas devem ser fornecidas quando um programa é checado.

DEFINIÇÃO 1 Diz-se que um programa, ou segmento de programa, S está *parcialmente correto com relação* à asserção inicial p e à asserção final q se sempre que p for verdadeira para os valores de entrada de S e S terminar, então q é verdadeira para os valores de saída de S. A notação $p\{S\}q$ indica que o programa, ou segmento de programa, S está parcialmente correto em relação à asserção inicial p e à asserção final q.

Nota: A notação $p\{S\}q$ é conhecida como *tripla de Hoare*. Tony Hoare introduziu o conceito de exatidão – ou corretude – parcial.

Links

Note que a noção de exatidão parcial não tem nada a ver com o fato de o programa terminar ou não; ela busca apenas saber se o programa faz o que é esperado que ele faça se ele terminar.

Um simples exemplo ilustra os conceitos de asserções inicial e final.

EXEMPLO 1 Mostre que o segmento de programa

$y := 2$
$z := x + y$

Exemplos Extras

está correto em relação à asserção inicial p: $x = 1$ e à asserção final q: $z = 3$.

Solução: Suponha que p seja verdadeira, de tal forma que $x = 1$ quando o programa começa. Então a y é designado o valor 2, e z é determinado pela soma dos valores de x e y, que é 3. Assim, S está correto com relação à asserção inicial p e à asserção final q. Então, $p\{S\}q$ é verdadeira. ◀

Regras de Inferência

Uma regra de inferência útil demonstra que um programa é correto ao separá-lo em uma série de subprogramas e, então, mostrando que cada subprograma está correto.

Suponha que o programa S tenha sido dividido nos subprogramas S_1 e S_2. Escreva $S = S_1; S_2$ para indicar que S é constituído por S_1 seguido de S_2. Suponha que a exatidão de S_1 com relação à asserção inicial p e à final q, e a exatidão de S_2 com relação à asserção inicial q e à final r tenham sido estabelecidas. Temos que, se p for verdadeira e S_1 for executado e terminar, então q é verdadeira; e se q for verdadeira e S_2 for executado e terminar, então r também é verdadeira. Assim, se p for verdadeira e $S = S_1; S_2$ for executado e terminar, então r é verdadeira. Essa regra de inferência, chamada de **regra de composição**, pode ser estabelecida como

$$\begin{array}{l} p\{S_1\}q \\ \underline{q\{S_2\}r} \\ \therefore\ p\{S_1;S_2\}r. \end{array}$$

Essa regra de inferência será usada mais adiante nesta seção.

A seguir, veremos algumas regras de inferência para segmentos de programas que envolvem proposições condicionais e laços. Como os programas podem ser divididos em segmentos para as demonstrações de exatidão, isso nos permitirá verificar muitos programas diferentes.

Proposições Condicionais

Primeiro serão dadas as regras de inferência para as proposições condicionais. Suponha que um segmento de programa tenha a forma

```
if condicao then
    S
```

em que S é um bloco de proposições. Então, S é executado se a *condição* for verdadeira e não será executado quando a *condição* for falsa. Para verificar se esse segmento está correto em relação à asserção inicial p e à final q, duas coisas devem ser feitas. Primeiro, deve ser mostrado que, quando p é verdadeira e a *condição* também é verdadeira, então q é verdadeira depois que S terminar. Depois devemos mostrar que, quando p é verdadeira e a *condição* é falsa, então q é verdadeira (porque neste caso S não é executado).

Isso nos leva à seguinte regra de inferência:

$$\begin{array}{l} (p \wedge condicao)\{S\}q \\ \underline{(p \wedge \neg condicao) \rightarrow q} \\ \therefore\ p\{\textbf{if } condicao \textbf{ then } S\}q. \end{array}$$

O Exemplo 2 ilustra como esta regra de inferência é usada.

Links

C. ANTHONY R. HOARE (Nascido em 1934) Tony Hoare é atualmente Professor de Ciência da Computação na Universidade de Oxford, na Inglaterra, e é Membro da Royal Society. Hoare fez muitas contribuições importantes para a teoria da linguagem de programação e para a metodologia de programação. Ele foi o primeiro a definir uma linguagem de programa com base em como seria possível demonstrar que os programas estavam corretos com relação a suas especificações. Hoare é também o criador do quick sort, um dos algoritmos de ordenação mais comumente usados e estudados (veja o preâmbulo do Exercício 50 da Seção 4.4). Hoare é um escritor notável nos aspectos técnicos e sociais da ciência da computação.

EXEMPLO 2 Verifique se o segmento de programa

```
if x > y then
    y := x
```

está correto com relação à sua asserção inicial **V** e à sua asserção final $y \geq x$.

Solução: Quando a asserção inicial for verdadeira e $x > y$, a tarefa $y := x$ é realizada. Assim, a asserção final, que determina que $y \geq x$, é verdadeira neste caso. Além disso, quando a asserção inicial for verdadeira e $x > y$ for falsa, então ocorre $x \leq y$, a asserção final é novamente verdadeira. Assim, usando a regra de inferência para segmentos de programa desse tipo, este programa está correto com relação às suas asserções inicial e final. ◀

Da mesma forma, suponha que um programa tenha o seguinte enunciado

```
if condicao then
    S1
else
    S2
```

Se a *condição* for verdadeira, então S_1 é executado; se a *condição* for falsa, então S_2 é executado. Para verificar se este segmento de programa está correto com relação à asserção inicial p e à asserção final q, duas coisas devem ser feitas. Primeiro, devemos mostrar que quando p for verdadeira e a *condição* for verdadeira, então q é verdadeira depois que S_1 terminar. Segundo, devemos mostrar que, quando p for verdadeira e a *condição* for falsa, então q é verdadeira depois que S_2 terminar. Isso nos leva à seguinte regra de inferência:

$$\frac{(p \wedge condicao)\{S_1\}q \qquad (p \wedge \neg condicao)\{S_2\}q}{\therefore\ p\{\textbf{if } condicao \textbf{ then } S_1 \textbf{ else } S_2\}q.}$$

O Exemplo 3 ilustra como esta regra de inferência é usada.

EXEMPLO 3 Verifique que o segmento de programa

```
if x < 0 then
    abs := -x
else
    abs := x
```

está correto com relação à asserção inicial **V** e à final $abs = |x|$.

Solução: Duas coisas devem ser demonstradas. Primeiro, devemos mostrar que a asserção inicial é verdadeira e $x < 0$, então $abs = |x|$. Isso é correto, porque, quando $x < 0$ a proposição $abs := -x$ é executada $abs = -x$, que é $|x|$ pela definição quando $x < 0$. Segundo, devemos mostrar que, se a asserção inicial for verdadeira e $x < 0$ for falso, então $abs = |x|$. Isso também é correto, porque neste caso o programa usa a proposição $abs := x$, e x é $|x|$ pela definição quando $x \geq 0$, então

abs := *x*. Assim, usando a regra de inferência para segmentos de programas deste tipo, este segmento está correto com relação às asserções dadas, inicial e final. ◀

Laços Invariáveis

Links

Agora, serão descritas as demonstrações de exatidão com laços **enquanto** (**while**). Para desenvolver uma regra de inferência para segmentos de programas do tipo

```
while condicao
    S
```

note que *S* é executado repetidamente até que a *condição* se torne falsa. Uma asserção que permanece verdadeira toda vez que *S* for executado deve ser escolhida. Essa asserção é chamada de **laço invariável.** Em outras palavras, *p* é um laço invariável se $(p \wedge condição)\{S\}p$ for verdadeira.

Suponha que *p* seja um laço invariável invariante. Temos que, se *p* for verdadeira antes de o segmento de programa ser executado, *p* e ¬*condição* são verdadeiros depois da finalização, se ela ocorrer. Esta regra de inferência é

$$\frac{(p \wedge condicao)\{S\}p}{\therefore\ p\{\textbf{while}\ condicao\ S\}(\neg\ condicao \wedge p).}$$

O uso de um laço invariável está ilustrado no Exemplo 4.

EXEMPLO 4 Um laço invariável é necessário para verificar que o segmento de programa

Exemplos Extras

```
i := 1
fatorial := 1
while i < n
begin
    i := i + 1
    fatorial := fatorial · i
end
```

termina com *fatorial* = *n*!, quando *n* for um número inteiro positivo.

Considere *p* como a asserção "*fatorial* = *i*! e $i \leq n$". Primeiro, demonstramos que *p* é um laço invariável. Suponha que, no começo de uma execução do laço **while**, *p* é verdadeira e a condição de laço **while** se mantém; em outras palavras, assuma que *fatorial* = *i*! e que $i < n$. Os novos valores i_{novo} e $fatorial_{novo}$ de *i* e *fatorial* são $i_{novo} = i + 1$ e $fatorial_{novo} = fatorial \cdot (i + 1) = (i + 1)!$ $= i_{novo}!$. Como $i < n$, também temos que $i_{novo} = i + 1 \leq n$. Assim, *p* é verdadeira no final da execução do laço. Isso mostra que *p* é um laço invariável.

Agora, considere o segmento de programa. Apenas antes de entrar o laço, $i = 1 \leq n$ e *fatorial* $= 1 = 1! = i!$ se mantêm, assim, *p* é verdadeira. Como *p* é um laço invariável, a regra de inferência introduzida implica que se o laço **while** terminar, ele termina com *p* verdadeira e com $i < n$ falso. Neste caso, no final, *fatorial* = *i*! e $i \leq n$ são verdadeiros, mas $i < n$ é falso; em outras palavras, $i = n$ e *fatorial* = *i*! = *n*!, como desejado.

Por fim, precisamos checar que o laço **while** termina de fato. No começo do programa, i é determinado com o valor 1, assim, depois de $n-1$ laços, o novo valor de i será n e o laço termina neste ponto. ◀

Um exemplo final será apresentado para mostrar como as várias regras de inferência podem ser usadas para verificar a exatidão de um programa maior.

EXEMPLO 5 Demonstraremos como verificar a exatidão do programa S para a computação do produto de dois números inteiros.

procedure *multiplicacao(m, n*: números inteiros)

S_1 **if** $n < 0$ **then** $a := -n$
else $a := n$

S_2 $k := 0$
$x := 0$

S_3 **while** $k < a$
begin
$x := x + m$
$k := k + 1$
end

S_4 **if** $n < 0$ **then** *produto* $:= -x$
else *produto* $:= x$

O objetivo é demonstrar que depois que S for executado, o *produto* tem valor mn. A demonstração da exatidão pode ser realizada dividindo S em quatro segmentos, com $S = S_1; S_2; S_3; S_4$, como mostrado anteriormente. A regra de composição pode ser usada para desenvolver a demonstração de exatidão. Aqui, mostramos como o argumento ocorre. Os detalhes serão deixados como um exercício para o leitor.

Considere p como a asserção inicial "m e n são números inteiros". Então, podemos mostrar que $p\{S_1\}q$ é verdadeira, quando q é a proposição $p \wedge (a = |n|)$. Depois, considere r como a proposição $q \wedge (k = 0) \wedge (x = 0)$. É fácil verificar que $q\{S_2\}r$ é verdadeira. Podemos mostrar que "$x = mk$ e $k \leq a$" é um laço invariável de S_3. Além disso, é fácil ver que o laço termina depois de a iterações, com $k = a$, assim $x = ma$ neste ponto. Como r implica que $x = m \cdot 0$ e $0 \leq a$, o laço invariável é verdadeiro antes que o laço seja inserido. Como o laço termina com $k = a$, temos que $r\{S_3\}s$ é verdadeira, em que s é a proposição "$x = ma$ e $a = |n|$". Por fim, podemos mostrar que S_4 está correto com relação à asserção inicial s e à final t, em que t é a proposição "*produto* $= mn$".

Juntando tudo, como $p\{S_1\}q$, $q\{S_2\}r$, $r\{S_3\}s$ e $s\{S_4\}t$ são todas verdadeiras, temos que, a partir da regra de composição, $p\{S\}t$ é verdadeira. Além disso, como todos os quatro segmentos finalizam, S também finaliza. Isso verifica a exatidão do programa. ◀

Exercícios

1. Demonstre que o segmento de programa

$y := 1$
$z := x + y$

está correto com relação à asserção inicial $x = 0$ e à final $z = 1$.

2. Verifique se o segmento de programa

if $x < 0$ **then** $x := 0$

está correto com relação à asserção inicial **V** e à final $x \geq 0$.

3. Verifique que o segmento de programa

```
x := 2
z := x + y
if y > 0 then
  z := z + 1
else
  z := 0
```

está correto com relação à asserção inicial $y = 3$ e à final $z = 6$.

4. Verifique que o segmento de programa

```
if x < y then
  min := x
else
  min := y
```

está correto com relação à asserção inicial **V** e à final $(x \leq y \wedge min = x) \wedge (x > y \wedge min = y)$.

*** 5.** Desenvolva uma regra de inferência para verificar a exatidão parcial das proposições na forma

```
if condicao 1 then
  S1
else if condicao 2 then
  S2
  ⋮
else
  Sn
```

em que $S_1, S_2, \ldots, S_n$ são blocos.

6. Use a regra de inferência desenvolvida no Exercício 5 para verificar se o programa

```
if x < 0 then
  y := −2|x|/x
else if x > 0 then
  y := 2|x|/x
else if x = 0 then
  y := 2
```

está correto com relação à asserção inicial **V** e à final $y = 2$.

7. Use um laço invariável para demonstrar que o seguimento de programa a seguir para a computação da n-ésima potência, em que n é um número inteiro positivo, de um número real x está correto.

```
potencia := 1
i := 1
while i ≤ n
begin
  potencia := potencia * x
  i := i + 1
end
```

***8.** Demonstre que o programa de iteração para encontrar f_n dado na Seção 4.4 é correto.

9. Forneça todos os detalhes da demonstração de exatidão dada no Exemplo 5.

10. Suponha que a proposição condicional $p_0 \rightarrow p_1$ e a asserção do programa $p_1\{S\}q$ sejam verdadeiras. Mostre que $p_0\{S\}q$ também deve ser verdadeira.

11. Suponha que a asserção do programa $p\{S\}q_0$ e a proposição condicional $q_0 \rightarrow q_1$ sejam verdadeiras. Mostre que $p\{S\}q_1$ também deve ser verdadeira.

12. O programa abaixo fará a computação dos quocientes e restos.

```
r := a
q := 0
while r ≥ d
begin
  r := r − d
  q := q + 1
end
```

Verifique que ele está parcialmente correto com relação à asserção inicial "a e d são números inteiros positivos" e à final "q e r são números inteiros, tal que $a = dq + r$ e $0 \leq r < d$".

13. Use um laço invariável para verificar que o algoritmo de Euclides (Algoritmo 6 da Seção 3.6) está parcialmente correto com relação à asserção inicial "a e b são números inteiros positivos" e à final "$x = \text{mdc}(a, b)$".

Termos-chave e Resultados

TERMOS

seqüência: uma função com domínio que é um subconjunto do conjunto dos números inteiros.

progressão geométrica: uma seqüência na forma $a, ar, ar^2, \ldots$, em que a e r são números reais.

progressão aritmética: uma seqüência na forma $a, a + d, a + 2d, \ldots$, em que a e d são números reais.

o princípio da indução matemática: a proposição $\forall n\, P(n)$ é verdadeira se $P(1)$ for verdadeira e se $\forall k[P(k) \rightarrow P(k + 1)]$ for verdadeira.

passo base: a demonstração de $P(1)$ em uma demonstração por indução matemática de $\forall nP(n)$.

passo de indução: a demonstração de $P(k) \rightarrow P(k + 1)$ para todos os números inteiros positivos k em uma demonstração por indução matemática de $\forall nP(n)$.

indução completa (ou indução forte): a proposição $\forall nP(n)$ é verdadeira se $P(1)$ for verdadeira e se $\forall k[(P(1) \wedge \cdots \wedge P(k)) \rightarrow P(k + 1)]$ for verdadeira.

propriedade da boa ordenação: todo conjunto não vazio de números inteiros não negativos tem pelo menos um elemento.

definição recursiva de uma função: a definição de uma função que especifica um conjunto inicial de valores e uma regra para obter valores dessa função com números inteiros a partir de seus valores em números inteiros menores.

definição recursiva de um conjunto: a definição de um conjunto que especifica um conjunto inicial de elementos em um conjunto e uma regra para obter outros elementos a partir daqueles no conjunto.

indução estrutural: a técnica para demonstrar resultados de conjuntos definidos recursivamente.

algoritmo recursivo: o algoritmo que funciona reduzindo um problema ao mesmo problema, só que com entradas menores.

merge sort: o algoritmo de ordenação que organiza uma lista separando-a em duas, ordenando cada uma das duas listas resultantes e unindo os resultados em uma lista ordenada.

iteração: o procedimento que se baseia no uso da repetição das operações em um laço.

exatidão de programas: a verificação de que um procedimento sempre produz um resultado correto.

laço invariável: a propriedade que permanece verdadeira durante cada passagem por comandos de um laço.

asserção inicial: a proposição que especifica as propriedades dos valores de entrada de um programa.

asserção final: a proposição que especifica as propriedades que os valores de saída devem ter se o programa funcionar corretamente.

Questões de Revisão

1. **a)** Você pode usar o princípio da indução matemática a fim de encontrar uma fórmula para a soma dos primeiros n termos de uma seqüência?
b) Você pode usar o princípio da indução matemática a fim de determinar se uma fórmula dada para a soma dos primeiros n termos de uma seqüência está correta?
c) Encontre uma fórmula para a soma dos primeiros n números inteiros positivos e pares e demonstre-a usando a indução matemática.

2. **a)** Para quais números inteiros positivos n é $11n + 17 \leq 2^n$?
b) Demonstre a conjectura que você formulou no item (a) usando a indução matemática.

3. **a)** Quais valores de postagem podem ser formados usando-se apenas selos de 5 e 9 centavos?
b) Demonstre a conjectura que você formulou usando a indução matemática.
c) Demonstre a conjectura que você formulou usando a indução completa.
d) Encontre uma demonstração para a sua conjectura que seja diferente daquelas que você construiu nos itens (b) e (c).

4. Dê dois exemplos diferentes de demonstrações que usem a indução completa.

5. **a)** Estabeleça a propriedade da boa ordenação para o conjunto dos números inteiros positivos.
b) Use essa propriedade para mostrar que todo número inteiro positivo pode ser escrito como o produto de números primos.

6. **a)** Explique por que uma função é bem definida se ela for definida recursivamente, especificando $f(1)$ e uma regra para encontrar $f(n)$ a partir de $f(n-1)$.
b) Forneça uma definição recursiva da função $f(n) = (n+1)!$.

7. **a)** Dê uma definição recursiva dos números de Fibonacci.
b) Mostre que $f_n > \alpha^{n-2}$ sempre que $n \geq 3$, em que f_n é o n-ésimo termo da seqüência de Fibonacci e $\alpha = (1+\sqrt{5})/2$.

8. **a)** Explique por que uma seqüência a_n é bem definida se ela for definida recursivamente especificando a_1 e a_2 e uma regra para encontrar a_n a partir de $a_1, a_2, \ldots, a_{n-1}$ para $n = 3, 4, 5, \ldots$.
b) Encontre o valor de a_n se $a_1 = 1$, $a_2 = 2$ e $a_n = a_{n-1} + a_{n-2} + \cdots + a_1$, para $n = 3, 4, 5, \ldots$.

9. Dê dois exemplos de como as fórmulas bem formadas são definidas recursivamente para diferentes conjuntos de elementos e operadores.

10. **a)** Dê uma definição recursiva da extensão de uma cadeia.
b) Use a definição recursiva do item (a) e a indução estrutural para demonstrar que $l(xy) = l(x) + l(y)$.

11. **a)** O que é um algoritmo recursivo?
b) Descreva um algoritmo recursivo para a computação da soma dos n números em uma seqüência.

12. Descreva um algoritmo recursivo para a computação do máximo divisor comum de dois números inteiros positivos.

13. **a)** Descreva o algoritmo merge sort.
b) Use o algoritmo merge sort para colocar a lista 4, 10, 1, 5, 3, 8, 7, 2, 6, 9 em ordem crescente.
c) Dê uma estimativa O grande para o número de comparações usadas pela merge sort.

14. **a)** Testar um programa de computação para ver se ele produz uma saída correta para determinados valores de entrada verifica que o programa sempre produz as saídas corretas?
b) Mostrar que um programa de computação é parcialmente correto em relação à asserção inicial e à final verifica que ele sempre produz as saídas corretas? Se não, o que mais é necessário?

15. Quais técnicas você pode usar para mostrar que um programa de computação longo é parcialmente correto em relação à asserção inicial e à final?

16. O que é um laço invariável? Como ele é usado?

Exercícios Complementares

1. Use a indução matemática para mostrar que $\frac{2}{3} + \frac{2}{9} + \frac{2}{27} + \cdots + \frac{2}{3^n} = 1 - \frac{1}{3^n}$ sempre que n for um número inteiro positivo.

2. Use a indução matemática para mostrar que $1^3 + 3^3 + 5^3 + \cdots + (2n+1)^3 + (n+1)^2(2n^2 + 4n + 1)$ sempre que n for um número inteiro positivo.

3. Use a indução matemática para mostrar que $1 \cdot 2^0 + 2 \cdot 2^1 + 3 \cdot 2^2 + \cdots + n \cdot 2^{n-1} = (n-1) \cdot 2^n + 1$ sempre que n for um número inteiro positivo.

4. Use a indução matemática para mostrar que

$$\frac{1}{1 \cdot 3} + \frac{1}{3 \cdot 5} + \ldots + \frac{1}{(2n-1)(2n+1)} = \frac{n}{2n+1}$$

sempre que n for um número inteiro positivo.

5. Mostre que

$$\frac{1}{1 \cdot 4} + \frac{1}{4 \cdot 7} + \ldots + \frac{1}{(3n-2)(3n+1)} = \frac{n}{3n+1}$$

sempre que n for um número inteiro positivo.

6. Use a indução matemática para demonstrar que $2^n > n^2 + n$ sempre que n for um número inteiro maior que 4.

7. Use a indução matemática para mostrar que $2^n > n^3$ sempre que n for um número inteiro maior que 9.

8. Encontre um número inteiro N, tal que $2^n > n^4$ sempre que n for maior que N. Demonstre que seu resultado está correto usando a indução matemática.

9. Use a indução matemática para demonstrar que $a - b$ é um fator de $a^n - b^n$ sempre que n for um número inteiro positivo.

10. Use a indução matemática para demonstrar que 9 divide $n^3 + (n+1)^3 + (n+2)^3$ sempre que n for um número inteiro não negativo.

11. Use a indução matemática para demonstrar esta fórmula para a soma dos termos de uma progressão aritmética.

$$a + (a+d) + \cdots + (a+nd) = (n+1)(2a+nd)/2$$

12. Suponha que $a_j \equiv b_j \pmod{m}$ para $j = 1, 2, \ldots, n$. Use a indução matemática para demonstrar que

a) $\sum_{j=1}^{n} a_j \equiv \sum_{j=1}^{n} b_j \pmod{m}$.

b) $\prod_{j=1}^{n} a_j \equiv \prod_{j=1}^{n} b_j \pmod{m}$.

13. Mostre que se n for um número inteiro positivo, então

$$\sum_{k=1}^{n} \frac{k+4}{k(k+1)(k+2)} = \frac{n(3n+7)}{2(n+1)(n+2)}.$$

14. Para quais números inteiros positivos n é $n + 6 < (n^2 - 8n)/16$? Demonstre sua resposta usando a indução matemática.

15. (*Requer cálculo*) Suponha que $f(x) = e^x$ e $g(x) = xe^x$. Use a indução matemática juntamente com a regra do produto e o fato de que $f'(x) = e^x$ para demonstrar que $g^{(n)}(x) = (x+n)e^x$ sempre que n for um número inteiro positivo.

16. (*Requer cálculo*) Suponha que $f(x) = e^x$ e $g(x) = e^{cx}$, em que c é uma constante. Use a indução matemática juntamente com a regra da cadeia e o fato de que $f'(x) = e^x$ para demonstrar que $g^{(n)} = c^n e^{cx}$ sempre que n for um número inteiro positivo.

***17.** Determine quais números de Fibonacci são pares e use uma forma da indução matemática para demonstrar sua conjectura.

***18.** Determine quais números de Fibonacci são divisíveis por 3. Use uma forma da indução matemática para demonstrar sua conjectura.

***19.** Demonstre que $f_k f_n + f_{k+1} f_{n+1} = f_{n+k+1}$ para todos os números inteiros não negativos n, em que k é um número inteiro não negativo e f_i indica o i-ésimo número de Fibonacci.

A seqüência dos **números de Lucas** é definida por $l_0 = 2$, $l_1 = 1$ e $l_n = l_{n-1} + l_{n-2}$ para $n = 2, 3, 4, \ldots$.

20. Mostre que $f_n + f_{n+2} = l_{n+1}$ sempre que n for um número inteiro positivo, em que f_i e l_i são o i-ésimo número de Fibonacci e o i-ésimo número de Lucas, respectivamente.

21. Mostre que $l_0^2 + l_1^2 + \ldots + l_n^2 = l_n l_{n+1} + 2$ sempre que n for um número inteiro não negativo e l_i for o i-ésimo número de Lucas.

***22.** Use a indução matemática para mostrar que o produto de quaisquer n números inteiros positivos consecutivos é divisível por $n!$. [*Dica:* Use a identidade $m(m+1)\cdots(m+n-1)/n! = (m-1)m(m+1)\cdots(m+n-2)/n! + m(m+1)\cdots(m+n-2)/(n-1)!$.]

23. Use a indução matemática para mostrar que $(\cos x + i \operatorname{sen} x)^n = \cos nx + i \operatorname{sen} nx$ sempre que n for um número inteiro positivo. (Aqui, i é a raiz quadrada de -1.) [*Dica:* Use a identidade $\cos(a+b) = \cos a \cos b - \operatorname{sen} a \operatorname{sen} b$ e $\operatorname{sen}(a+b) = \operatorname{sen} a \cos b + \cos a \operatorname{sen} b$.]

***24.** Use a indução matemática para mostrar que $\sum_{j=1}^{n} \cos jx = \cos[(n+1)x/2] \operatorname{sen}(nx/2)/\operatorname{sen}(x/2)$ sempre que n for um número inteiro positivo e $\operatorname{sen}(x/2) \neq 0$.

25. Use a indução matemática para demonstrar que $\sum_{j=1}^{n} j^2 2^j = n^2 2^{n+1} - n2^{n+2} + 3 \cdot 2^{n+1} - 6$ para todo número inteiro positivo n.

26. (*Requer cálculo*) Suponha que a seqüência $x_1, x_2, \ldots, x_n, \ldots$ é definida recursivamente por $x_1 = 0$ e $x_{n+1} = \sqrt{x_n + 6}$.

a) Use a indução matemática para mostrar que $x_1 < x_2 < \cdots < x_n < \cdots$, ou seja, a seqüência $\{x_n\}$ é monotonicamente crescente.

b) Use a indução matemática para demonstrar que $x_n < 3$ para $n = 1, 2, \ldots$.

c) Mostre que $\lim_{n \to \infty} x_n = 3$.

27. Mostre que se n for um número inteiro positivo com $n \geq 2$, então

$$\sum_{j=2}^{n} \frac{1}{j^2 - 1} = \frac{(n-1)(3n+2)}{4n(n+1)}.$$

28. Use a indução matemática para demonstrar o Teorema 1 da Seção 3.6, ou seja, mostre que se b for um número inteiro positivo, $b > 1$, e n for um número inteiro positivo, então n pode ser expresso unicamente na forma $n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$.

***29.** Um **ponto reticulado** no plano é um ponto (x, y) em que x e y são números inteiros. Use a indução matemática para mostrar que pelo menos $n+1$ linhas retas são necessárias para garantir que todo ponto reticulado (x, y) com $x \geq 0$, $y \geq 0$ e $x + y \leq n$ pertencem a essas linhas retas.

30. (*Requer cálculo*) Use a indução matemática e a regra do produto para mostrar que se n for um número inteiro positivo e $f_1(x)$, $f_2(x)$, ..., $f_n(x)$ forem funções diferenciáveis, então

$$\frac{(f_1(x)f_2(x)\cdots f_n(x))'}{f_1(x)f_2(x)\cdots f_n(x)} = \frac{f_1'(x)}{f_1(x)} + \frac{f_2'(x)}{f_2(x)} + \ldots + \frac{f_n'(x)}{f_n(x)}.$$

31. (*Requer matéria da Seção 3.7*) Suponha que $\mathbf{B} = \mathbf{MAM}^{-1}$, em que $\boldsymbol{A}$ e $\boldsymbol{B}$ são matrizes $n \times n$ e $\mathbf{M}$ é inversível. Mostre que $\mathbf{B}^k = \mathbf{MA}^k\mathbf{M}^{-1}$ para todos os números inteiros positivos k.

32. Use a indução matemática para mostrar que se você desenhar linhas no plano, precisará apenas de duas cores para colorir as regiões formadas, para que duas regiões que têm uma fronteira comum não apresentem a mesma cor.

33. Mostre que $n!$ pode ser representado como a soma de n de seus divisores positivos distintos, sempre que $n \geq 3$. [*Dica:* Use a carga indutiva. Primeiro tente demonstrar este resultado usando a indução matemática. Examinando onde sua demonstração falhou, encontre uma proposição forte que você pode demonstrar facilmente usando a indução matemática.]

***34.** Use a indução matemática para demonstrar que se $x_1, x_2, \ldots, x_n$ forem números reais positivos com $n \geq 2$, então

$$\left(x_1 + \frac{1}{x_1}\right)\left(x_2 + \frac{1}{x_2}\right)\cdots\left(x_n + \frac{1}{x_n}\right) \geq \left(x_1 + \frac{1}{x_2}\right)\left(x_2 + \frac{1}{x_3}\right)\cdots\left(x_{n-1} + \frac{1}{x_n}\right)\left(x_n + \frac{1}{x_1}\right)$$

35. Use a indução matemática para demonstrar que se n pessoas estiverem paradas em uma fila, em que n é um número inteiro positivo, e se a primeira pessoa da fila for uma mulher e a última for um homem, então em algum lugar da fila há uma mulher diretamente na frente de um homem.

***36.** Suponha que para cada par de cidades em um país, há uma estrada direta conectando-as em uma direção ou na outra. Use a indução matemática para mostrar que há uma cidade que pode ser alcançada a partir de qualquer outra cidade ou diretamente ou através de outra cidade.

37. Use a indução matemática para mostrar que quando n círculos dividem o plano em regiões, essas regiões podem ser coloridas com duas cores diferentes, tal que nenhuma região com margens comuns sejam coloridas com a mesma cor.

***38.** Suponha que em um grupo de carros em uma pista circular exista combustível suficiente para um carro completar a volta. Use a indução matemática para mostrar que há um carro no grupo que pode completar uma volta obtendo o combustível dos carros enquanto ele atravessa a pista.

39. Mostre que se n for um número inteiro positivo, então

$$\sum_{j=1}^{n}(2j-1)\left(\sum_{k=j}^{n}1/k\right) = n(n+1)/2.$$

40. Uma **unidade** ou **fração egípcia** é uma fração na forma $1/n$, em que n é um número inteiro positivo. Neste exercício, usaremos a indução completa para mostrar que um algoritmo "voraz" pode ser usado para expressar qualquer número racional p/q com $0 < p/q < 1$ como a soma de unidades de frações distintas. Em cada passo do algoritmo, encontramos o menor número inteiro positivo n, tal que $1/n$ possa ser adicionado à soma sem exceder p/q. Por exemplo, para expressar $5/7$, primeiro começamos somando com $1/2$. Como $5/7 - 1/2 = 3/14$, adicionamos $1/5$ à soma porque 5 é o menor número inteiro positivo k, tal que $1/k < 3/14$. Como $3/14 - 1/5 = 1/70$, o algoritmo termina, mostrando que $5/7 = 1/2 + 1/5 + 1/70$. Considere $T(p)$ como a proposição que afirma que este algoritmo termina para todos os números racionais p/q com $0 < p/q < 1$. Vamos demonstrar que o algoritmo sempre termina mostrando que $T(p)$ é válida para todos os números inteiros positivos p.

a) Mostre que o passo base $T(1)$ é válido.

b) Suponha que se $T(k)$ é válida para os números inteiros positivos k com $k < p$, ou seja, assuma que o algoritmo termina para todos os números racionais k/r, em que $1 \leq k < p$. Mostre que se começarmos com p/q e a fração $1/n$ for selecionada no primeiro passo do algoritmo, então $p/q = p'/q' + 1/n$, em que $p' = np - q$ e $q' = nq$. Depois de considerar o caso em que $p/q = 1/n$, use a hipótese indutiva para mostrar que o algoritmo voraz termina quando começar com p'/q' e complete o passo de indução.

A **função 91 de McCarthy** (definida por John McCarthy, um dos fundadores da inteligência artificial) é definida usando-se a regra

$$M(n) = \begin{cases} n - 10 & \text{se } n > 100 \\ M(M(n+11)) & \text{se } n \leq 100 \end{cases}$$

para todos os números inteiros positivos n.

41. Usando sucessivamente a regra definida por $M(n)$, encontre

a) $M(102)$. **b)** $M(101)$. **c)** $M(99)$.

d) $M(97)$. **e)** $M(87)$. **f)** $M(76)$.

****42.** Mostre que a função $M(n)$ é uma função bem definida a partir do conjunto dos números inteiros positivos para o conjunto dos números inteiros positivos. [*Dica:* Demonstre que $M(n) = 91$ para todos os números inteiros positivos n com $n \leq 101$.]

43. A demonstração a seguir para

$$\frac{1}{1\cdot 2} + \frac{1}{2\cdot 3} + \ldots + \frac{1}{(n-1)n} = \frac{3}{2} - \frac{1}{n},$$

sempre que n for um número inteiro positivo é correta? Justifique sua resposta.

Passo base: O resultado é verdadeiro quando $n = 1$ porque

$$\frac{1}{1\cdot 2} = \frac{3}{2} - \frac{1}{1}.$$

Passo de indução: Assuma que o resultado seja verdadeiro para n. Então,

$$\frac{1}{1\cdot 2} + \frac{1}{2\cdot 3} + \ldots + \frac{1}{(n-1)n} + \frac{1}{n(n+1)} = \frac{3}{2} - \frac{1}{n} + \left(\frac{1}{n} - \frac{1}{n+1}\right) = \frac{3}{2} - \frac{1}{n+1}.$$

Assim, o resultado é verdadeiro para $n + 1$ se for verdadeiro para n. Isso completa a demonstração.

44. Suponha que $A_1, A_2, \ldots, A_n$ seja um grupo de conjuntos. Suponha que $R_2 = A_1 \oplus A_2$ e $R_k = R_{k-1} \oplus A_k$ para $k = 3, 4, \ldots, n$. Use a indução matemática para demonstrar que $x \in R_n$ se e somente se x pertencer a um número ímpar de conjuntos $A_1, A_2, \ldots, A_n$. (Lembre-se de que $S \oplus T$ é a diferença simétrica dos conjuntos S e T definida na Seção 2.2.)

***45.** Mostre que n círculos dividem o plano em $n^2 - n + 2$ regiões se para cada dois círculos eles se interceptam exatamente em dois pontos e se não houver três círculos com pontos em comum.

***46.** Mostre que n planos dividem um espaço tridimensional em $(n^3 + 5n + 6)/6$ regiões se cada três desses planos tiverem um ponto em comum e não houver quatro planos com um ponto em comum.

***47.** Use a propriedade da boa ordenação para mostrar que $\sqrt{2}$ é irracional. [*Dica:* Assuma que $\sqrt{2}$ é irracional. Mostre que o conjunto dos números inteiros positivos na forma $b\sqrt{2}$ tem pelo menos um elemento a. Então, mostre que $a\sqrt{2} - a$ é o menor número inteiro positivo nesta forma.]

48. Um conjunto é **bem ordenado** se todo subconjunto não vazio desse conjunto tiver um menor elemento. Determine se cada um dos conjuntos a seguir é bem ordenado:

a) o conjunto dos números inteiros.

b) o conjunto dos números inteiros maiores que -100.

c) o conjunto dos números racionais positivos.

d) o conjunto dos números racionais positivos com denominador menor que 100.

49. a) Mostre que se $a_1, a_2, \ldots, a_n$ forem números inteiros positivos, então $\text{mdc}(a_1, a_2, \ldots, a_{n-1}, a_n) = \text{mdc}(a_1, a_2, \ldots, a_{n-2}, \text{mdc}(a_{n-1}, a_n))$.

b) Use o item (a), juntamente com o algoritmo de Euclides, para desenvolver um algoritmo recursivo para computar o máximo divisor comum de um conjunto com n números inteiros positivos.

***50.** Descreva um algoritmo recursivo para escrever o máximo divisor comum de n números inteiros positivos com uma combinação linear desses inteiros.

51. Encontre uma fórmula explícita para $f(n)$ se $f(1) = 1$ e $f(n) = f(n-1) + 2n - 1$ para $n \geq 2$. Demonstre seu resultado usando a indução matemática.

****52.** Dê uma definição recursiva do conjunto de cadeias de bits que contenham duas vezes mais 0 que 1.

53. Considere S como o conjunto de cadeias de bits definidas recursivamente por $\lambda \in S$ e $0x \in S$, $x1 \in S$ se $x \in S$, em que λ é a cadeia vazia.

a) Encontre todas as cadeias em S de extensão não excedente a cinco.

b) Dê uma descrição explícita dos elementos de S.

54. Considere S como o conjunto de cadeias definidas recursivamente por $abc \in S$, $bac \in S$ e $acb \in S$, em que a, b e c são letras fixas; e para todo $x \in S$, $abcx \in S$; $abxc \in S$, $axbc \in S$ e $xabc \in S$, em que x é uma variável que representa uma cadeia de letras.

a) Encontre todos os elementos de S de extensão oito ou menos.

b) Mostre que todo elemento de S tem uma extensão divisível por três.

O conjunto B para todas as **cadeias balanceadas de parênteses** é definido recursivamente por $\lambda \in B$, em que λ é a cadeia vazia; $(x) \in B$, $xy \in B$ se $x, y \in B$.

55. Mostre que (( )( )) é uma cadeia balanceada de parênteses e (( ))) não é uma cadeia balanceada de parênteses.

56. Encontre todas as cadeias balanceadas de parênteses com exatamente seis símbolos.

57. Encontre todas as cadeias balanceadas de parênteses com quatro símbolos ou menos.

58. Use a indução para mostrar que se x for uma cadeia balanceada de parênteses, então o número de parênteses à esquerda é igual ao número de parênteses à direita em x. Defina a função N no conjunto de cadeias de parênteses por

$$N(\lambda) = 0, \quad N((\,) = 1, \quad N(\,)) = -1,$$
$$N(uv) = N(u) + N(v),$$

em que λ é a cadeia vazia e u e v são cadeias. Podemos mostrar que N é bem definida.

Links

JOHN McCARTHY (Nascido em 1927) John McCarthy nasceu em Boston. Ele cresceu em Boston e em Los Angeles. Foi estudante de graduação e mestrado de matemática e recebeu seu título de mestre em 1948 pelo Instituto de Tecnologia da Califórnia e seu Ph.D. em 1951, em Princeton. Depois de se graduar em Princeton, McCarthy ocupou cargos em Princeton, Stanford, Dartmouth e no M.I.T. Ele ocupou um cargo na Universidade de Stanford de 1962 a 1994, onde é agora professor emérito. Em Stanford, ele foi diretor do Laboratório de Inteligência Artificial, e ocupou uma cadeira na Escola de Engenharia, sendo um membro de Instituto Hoover.

McCarthy foi um pioneiro no estudo da inteligência artificial, termo que ele cunhou em 1955. Trabalhou com problemas relacionados a raciocínios e informações necessários para o comportamento de um computador inteligente. McCarthy foi o primeiro, dentre muitos cientistas da computação, a criar sistemas de computação do tipo *time sharing*. Ele desenvolveu a LISP, uma linguagem de computação que usa expressões simbólicas. Desempenhou um papel importante usando a lógica para verificar a exatidão de programas computacionais. McCarthy também trabalhou com as implicações sociais da tecnologia computacional. Atualmente trabalha com o problema de como as pessoas e os computadores fazem conjecturas por meio de hipóteses de que as complicações são ausentes das situações.

McCarthy defende a sustentabilidade do progresso humano e é otimista quanto ao futuro da humanidade. Ele também começou a escrever ficção científica. Alguns de seus trabalhos recentes exploram a possibilidade de o mundo ser um programa de computação escrito por alguma força maior. Dentre os prêmios que McCarthy ganhou, podemos citar o Prêmio Turing da Associação para Maquinário Computacional, o Prêmio de Excelência em Pesquisa da Conferência Internacional de Inteligência Artificial, o Prêmio Kyoto e a Medalha Nacional de Ciência.

59. Encontre
- **a)** $N(( ))$.
- **b)** $N( ))( ))(( )$.
- **c)** $N((( )(( ))$.
- **d)** $N(( )((( ))(( )))$.

****60.** Mostre que uma cadeia w de parênteses é balanceada se e somente se $N(w) = 0$ e $N(u) \geq 0$ sempre que u for uma parte inicial de w, ou seja, $w = uv$.

***61.** Dê um algoritmo recursivo para encontrar todas as cadeias balanceadas de parênteses que contêm o n símbolos ou menos.

62. Dê um algoritmo recursivo para encontrar $\text{mdc}(a, b)$, em que a e b são números inteiros não negativos, com base nestes fatos: $\text{mdc}(a, b) = \text{mdc}(b, a)$ se $a > b$, $\text{mdc}(0, b) = b$, $\text{mdc}(a, b) = 2\,\text{mdc}(a/2, b/2)$ se a e b forem pares, $\text{mdc}(a, b) = \text{mdc}(a/2, b)$ se a for par e b for ímpar e $\text{mdc}(a, b) = \text{mdc}(a, b - a)$.

63. Verifique o segmento de programa

if $x > y$ **then**

$x := y$

com relação à asserção inicial **V** e à final $x \leq y$.

***64.** Desenvolva uma regra de inferência para verificar programas recursivos e use-a para verificar o programa recursivo para computar fatoriais dados na Seção 4.4.

65. Desenvolva um algoritmo recursivo para contar o número de vezes que o número inteiro 0 aparece em uma lista de números inteiros.

Os exercícios 66 a 73 lidam com algumas seqüências incomuns, chamadas informalmente de **seqüências autogeradoras**, produzidas por relações de recorrência simples ou regras. Em particular, os exercícios 66 a 71 lidam com a seqüência $\{a(n)\}$ definida por $a(n) = n - a(a(n-1))$ para $n \geq 1$ e $a(0) = 0$. (Esta seqüência, assim como aquelas dos exercícios 70 e 71, são definidas no fascinante livro de Douglas Hofstader, *Gödel, Escher, Bach* ([Ho99]).

66. Encontre os primeiros 10 termos da seqüência $\{a(n)\}$ definida no preâmbulo deste exercício.

***67.** Demonstre que esta seqüência é bem definida, ou seja, mostre que $a(n)$ é definido unicamente para todos os números inteiros não negativos n.

****68.** Demonstre que $a(n) = \lfloor (n+1)\mu \rfloor$ em que $\mu = (-1 + \sqrt{5})/2$. [*Dica:* Primeiro mostre para todo $n > 0$ que $(\mu n - \lfloor \mu n \rfloor) + (\mu^2 n - \lfloor \mu^2 n \rfloor) = 1$. Então, mostre para todos os números reais α com $0 \leq \alpha < 1$ e $\alpha \neq 1 - \mu$ que $\lfloor (1+\mu)(1-\alpha) \rfloor + \lfloor \alpha + \mu \rfloor = 1$, considerando os casos $0 \leq \alpha < 1 - \mu$ e $1 - \mu < \alpha < 1$ separadamente.]

***69.** Use a fórmula do Exercício 68 para mostrar que $a(n) = a(n-1)$ se $\mu n - \lfloor \mu n \rfloor < 1 - \mu$ e $a(n) = a(n-1) + 1$.

70. Encontre os primeiros 10 termos de cada uma das seguintes seqüências autogeradoras:
- **a)** $a(n) = n - a(a(a(n-1)))$ para $n \geq 1$ com $a(0) = 0$
- **b)** $a(n) = n - a(a(a(a(n-1))))$ para $n \geq 1$ com $a(0) = 0$
- **c)** $a(n) = a(n - a(n-1)) + a(n - a(n-2))$ para $n \geq 3$ com $a(1) = 1$ e $a(2) = 1$

71. Encontre os primeiros 10 termos das seqüências $m(n)$ e $f(n)$ definidas pelo seguinte par de relações de recorrência entrelaçadas: $m(n) = n - f(m(n-1))$, $f(n) = n - m(f(n-1))$ para $n \geq 1$ com $f(0) = 1$ e $m(0) = 0$.

A **seqüência autogeradora de Golomb** é a única seqüência não decrescente de números inteiros positivos $a_1, a_2, a_3, \ldots$, que tem a propriedade de ter exatamente a_k ocorrências de k para cada número inteiro positivo k.

72. Encontre os primeiros 20 termos da seqüência autogeradora de Golomb.

***73.** Mostre que se $f(n)$ é o maior número inteiro m, tal que $a_m = n$, em que a_m é o m-ésimo termo da seqüência autogeradora de Golomb, então $f(n) = \sum_{k=1}^{n} a_k$ e $f(f(n)) = \sum_{k=1}^{n} k a_k$.

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

****1.** Dado um tabuleiro de damas $2^n \times 2^n$ com um quadrado faltando, construa o ladrilhamento deste tabuleiro usando triominós.

****2.** Construa todas as fórmulas bem formadas para as expressões que envolvem as variáveis x, y e z e os operadores $\{+, *, /, -\}$ com n símbolos ou menos.

****3.** Construa todas as fórmulas bem formadas para as proposições com n símbolos ou menos em que cada símbolo é **V**, **F**, uma das variáveis proposicionais p e q, ou um operador de $\{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$.

4. Dada uma cadeia, encontre sua reversa.

5. Dado um número real a e um número inteiro não negativo n, encontre a^n usando a recursividade.

6. Dado um número real a e um número inteiro não negativo n, encontre a^{2^n} usando a recursividade.

***7.** Dado um número real a e um número inteiro não negativo n, encontre a^n usando a expansão binária de n e um algoritmo recursivo para computar a^{2^k}.

8. Dados dois números inteiros diferentes de zero, encontre seu máximo divisor comum usando a recursividade.

9. Dada uma lista de números inteiros e um elemento x, localize x nesta lista usando uma implementação recursiva de uma busca linear.

10. Dada uma lista de números inteiros e um elemento x, localize x nessa lista usando uma implementação recursiva de uma busca binária.

11. Dado um número inteiro não negativo n, encontre o n-ésimo número de Fibonacci usando a iteração.

12. Dado um número inteiro não negativo n, encontre o n-ésimo número de Fibonacci usando a recursividade.

13. Dado um número inteiro positivo, encontre o número de partições deste inteiro. (Veja o Exercício 47 da Seção 4.3.)

14. Dados os números inteiros positivos m e n, encontre $A(m, n)$, o valor da função de Ackermann para o par (m, n). (Veja o preâmbulo do Exercício 48 da Seção 4.3.)

15. Dada uma lista de n números inteiros, ordene esses inteiros usando a merge sort.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Quais são os maiores valores de n para que $n!$ tenha menos que 100 dígitos decimais e menos que 1 000 dígitos decimais?

2. Determine quais números de Fibonacci são divisíveis por 5, quais são divisíveis por 7 e quais são divisíveis por 11. Demonstre que sua conjectura está correta.

3. Ladrilhe com triominós vários tabuleiros de damas 16×16, 32×32 e 64×64 com um quadrado faltando.

4. Explore quais tabuleiros de damas $m \times n$ podem ser completamente preenchidos por triominós. Você pode fazer uma conjectura que responda a esta questão?

****5.** Implemente um algoritmo para determinar se um ponto está no interior ou no exterior de um polígono simples.

****6.** Implemente um algoritmo para triangular um polígono simples.

7. Quais valores para a função de Ackermann são pequenos o suficiente para que você possa computá-los?

8. Compare o número de operações ou o tempo necessário para computar os números de Fibonacci recursivamente com aquele necessário para computá-los por iterações.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Descreva as origens da indução matemática. Quem foram as primeiras pessoas a usá-la e para quais problemas ela foi aplicada?

2. Explique como demonstrar o teorema da curva de Jordan para polígonos simples e descreva um algoritmo para determinar se um ponto está no interior ou no exterior de um polígono simples.

3. Descreva como a triangulação de polígonos simples é usada em alguns algoritmos-chave em geometria computacional.

4. Descreva várias aplicações diferentes dos números de Fibonacci para a biologia e as ciências físicas.

5. Discuta o uso da função de Ackermann na teoria das definições recursivas e na análise da complexidade de algoritmos para uniões de conjuntos.

6. Discuta algumas das várias metodologias usadas para estabelecer a exatidão de programas e compare-as com os métodos de Hoare descritos na Seção 4.5.

7. Explique como as idéias e os conceitos de exatidão de programas podem ser ampliados para demonstrar que os sistemas operacionais são seguros.

CAPÍTULO

5 Contagem

5.1 As Bases da Contagem
5.2 O Princípio da Casa dos Pombos
5.3 Permutações e Combinações
5.4 Coeficientes Binomiais
5.5 Permutações e Combinações Generalizadas
5.6 Gerando Permutações e Combinações

Combinatória, o estudo dos arranjos dos objetos, é uma parte importante da matemática discreta. Este assunto vem sendo estudado desde muitos anos antes do século XVII, quando questões combinatórias apareceram no estudo de jogos. A enumeração, contagem de objetos com certas propriedades, é uma parte importante da combinatória. Devemos contar os objetos para resolver muitos tipos diferentes de problemas. Por exemplo, a contagem é usada para determinar a complexidade de algoritmos. Ela também é exigida para determinar se há números de telefone ou endereços de protocolo da Internet suficientes para atender a demanda. Além disso, técnicas de contagem são extremamente usadas quando as probabilidades de eventos são computadas.

As regras básicas de contagem, que serão estudadas na Seção 5.1, podem resolver diversos problemas. Por exemplo, podemos usar essas regras para enumerar os diferentes números de telefone possíveis nos Estados Unidos, as senhas permitidas em um sistema computacional e as diferentes ordens nas quais os atletas podem terminar uma corrida. Outro importante instrumento combinatório é o princípio da casa dos pombos, que será estudado na Seção 5.2. Esse princípio afirma que quando os objetos são colocados em caixas e há mais objetos do que caixas, então há uma caixa que terá, pelo menos, 2 objetos. Por exemplo, podemos usar esse princípio para mostrar que no meio de um grupo de 15 estudantes ou mais, pelo menos 3 nasceram no mesmo dia da semana.

Podemos descrever muitos problemas de contagem em termos de arranjos com ou sem ordenação de objetos de um conjunto. Esses arranjos, chamados de permutação e combinação, são usados em muitos problemas de contagem. Por exemplo, suponha que os 100 finalistas de uma competição com 2000 estudantes foram convidados para um jantar. Podemos contar os grupos possíveis dos 100 estudantes que serão convidados, assim como o modo como os 10 primeiros podem ser premiados.

Outro problema de contagem envolve a criação de todos os arranjos de um tipo específico. Isto é freqüentemente usado em simulações computacionais. Imaginaremos algoritmos para gerar arranjos de diferentes tipos.

5.1 As Bases da Contagem

Introdução

Suponha que a senha de um sistema computacional possua seis, sete ou oito caracteres. Cada um desses caracteres deve ser um número ou uma letra do alfabeto. Cada senha deve conter no mínimo um número. Quantas senhas são possíveis? As técnicas necessárias para responder a essa pergunta e a vários outros problemas de contagem serão introduzidas nesta seção.

Os problemas de contagem aparecem em matemática e ciência da computação. Por exemplo, devemos contar as possibilidades favoráveis de um experimento e o total de possibilidades desse experimento para determinar a probabilidade dos eventos discretos. Precisamos contar o número de operações usado por um algoritmo para estudar seu tempo de complexidade.

Introduziremos as técnicas básicas de contagem nesta seção. Estes métodos servem como base para quase todas as técnicas de contagem.

Princípios Básicos de Contagem

Auto-avaliação

Apresentaremos dois princípios básicos de contagem, a **regra do produto** e a **regra da soma**. Depois mostraremos como elas podem ser usadas para resolver diferentes problemas de contagem.

A regra do produto aplica-se quando o procedimento é feito a partir de tarefas separadas.

REGRA DO PRODUTO Suponha que um procedimento possa ser dividido em uma seqüência de duas tarefas. Se houver n_1 formas de fazer a primeira tarefa e para cada uma dessas formas de fazer a primeira tarefa há n_2 formas de fazer a segunda tarefa, então há $n_1 n_2$ formas de concluir o procedimento.

Exemplos Extras Os exemplos de 1 a 10 mostram como a regra do produto é usada.

EXEMPLO 1 Uma nova companhia com apenas dois empregados, Sanchez e Patel, aluga um andar de um prédio com 12 salas. Quantas formas diferentes há para designar as diferentes salas para esses dois empregados?

Solução: O procedimento de distribuir as salas para os dois empregados consiste em atribuir uma sala para Sanchez, o que pode ser feito de 12 formas, e então atribuir uma sala a Patel, diferente da de Sanchez, o que pode ser feito de 11 formas. Pela regra do produto, há $12 \cdot 11 = 132$ formas de distribuir salas a esses dois empregados. ◀

EXEMPLO 2 As cadeiras de um auditório devem ser etiquetadas com uma letra e um número inteiro positivo que não exceda a 100. Qual é o maior número de cadeiras que podem ser etiquetadas de maneira diferente?

Solução: O procedimento de etiquetar uma cadeira consiste em duas tarefas, atribuir uma das 26 letras e depois atribuir um dos 100 possíveis números inteiros. A regra do produto mostra que há $26 \cdot 100 = 2\,600$ formas diferentes de etiquetar as cadeiras. Assim, o maior número de cadeiras que podem ser etiquetadas é 2 600. ◀

EXEMPLO 3 Há 32 computadores em um centro computacional. Cada computador tem 24 portas. Quantas portas diferentes para um computador existem no centro?

Solução: O procedimento de escolher uma porta consiste em duas tarefas, primeiro escolher um computador e depois escolher a porta nesse computador. Tendo 32 possibilidades para a escolha do computador e 24 possibilidades para a escolha da porta, não importando qual computador foi escolhido, a regra do produto mostra que há $32 \cdot 24 = 768$ portas. ◀

Uma versão expandida da regra do produto é muito útil. Suponha que um procedimento é desenvolvido a partir da realização das tarefas $T_1, T_2, \cdots, T_m$ em seqüência. Se cada tarefa T_i, $i = 1, 2, \cdots, n$, pode ser feita de n_i formas, não importando como as tarefas anteriores tenham sido realizadas, então há $n_1 \cdot n_2 \cdot \cdots \cdot n_m$ formas de realizar o procedimento. Esta versão da regra do produto pode ser demonstrada pela indução matemática da regra do produto para duas tarefas (veja o Exercício 60 no final da seção).

EXEMPLO 4 Quantas seqüências de bits com comprimento sete existem?

Solução: Cada um dos sete bits pode ser escolhido de duas formas, porque cada bit equivale a 0 ou 1. Assim, a regra do produto mostra que há no total $2^7 = 128$ diferentes seqüências de bits de comprimento sete. ◀

EXEMPLO 5 Quantas placas de identificação estão disponíveis, se cada placa contém uma seqüência de três letras seguidas de três números (e não são proibidas quaisquer seqüências de letras, mesmo se elas forem obscenas)?

Solução: Há 26 escolhas para cada uma das três letras e 10 escolhas para cada um dos três números. Pela regra do produto há um total de $26 \cdot 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 = 17\,576\,000$ possibilidades de placas de identificação. ◀

26 escolhas para cada letra; 10 escolhas para cada número

EXEMPLO 6 **Contando Funções** Quantas funções existem partindo-se de um conjunto com m elementos para um conjunto com n elementos?

Solução: Uma função corresponde a uma escolha de um dos n elementos no contradomínio para cada um dos m elementos no domínio. Portanto, pela regra do produto, há $n \cdot n \cdot \cdots \cdot n = n^m$ funções partindo-se de um conjunto com m elementos para um com n elementos. Por exemplo, há 5^3 funções diferentes partindo-se de um conjunto com três elementos para um conjunto com cinco elementos. ◀

EXEMPLO 7 **Contando Funções Injetoras** Quantas funções injetoras há partindo-se de um conjunto com m elementos para um com n elementos?

Solução: Primeiro note que quando $m > n$, não há funções injetoras ao se partir de um conjunto com m elementos para um com n elementos.

Agora, considere que $m \leq n$. Suponha que os elementos no domínio são $a_1, a_2, \cdots, a_m$. Há n formas de escolher o valor da função em a_1. Devido à função ser injetora, o valor da função em a_2 pode ser escolhido de $n - 1$ formas (porque o valor usado em a_1 não pode ser usado novamente). Geralmente, o valor da função em a_k pode ser escolhido de $n - k + 1$ formas. Pela regra do produto, há $n(n - 1)(n - 2) \cdots (n - m + 1)$ funções injetoras partindo-se de um conjunto com m elementos para um com n elementos.

Por exemplo, há $5 \cdot 4 \cdot 3 = 60$ funções injetoras partindo-se de um conjunto com três elementos para um com cinco elementos. ◀

EXEMPLO 8 **Prefixos de Telefone** O formato de números de telefone na América do Norte é especificado por um *prefixo telefônico*. Um número telefônico consiste em 10 dígitos, que são organizados em um código de área de três dígitos, um código de função de três dígitos e um código de localização de quatro dígitos. Por causa das considerações de sinal, há algumas restrições em alguns desses dígitos. Para especificar os formatos possíveis, considere X como um dígito que pode ser um valor de 0 a 9, N como um dígito que pode ter valor de 2 a 9 e Y como um dígito que deve ter um valor 0 ou 1. Dois prefixos telefônicos, que serão chamados de prefixo antigo e prefixo novo, serão discutidos. (O prefixo antigo, usado na década de 1960, foi substituído pelo novo, mas o recente crescimento na demanda por novos números de telefones móveis tornaram-no obsoleto. Neste exemplo, as letras usadas para representar os dígitos seguem as convenções do *Prefixo Telefônico da América do Norte.*) Como será mostrado, o novo prefixo permite o uso de mais números.

Links

No prefixo antigo, os formatos de código de área, código de função e código de localização são *NYX*, *NNX* e *XXXX*, respectivamente. Assim, os números de telefone tinham a forma *NYX-NNX-XXXX*. No novo prefixo, os formatos desses códigos são *NXX*, *NXX* e *XXXX*, respectivamente, então os números de telefone têm a forma *NXX-NXX-XXXX*. Quantos números diferentes no padrão norte-americano são possíveis no prefixo antigo e no prefixo novo?

Solução: Pela regra do produto, há $8 \cdot 2 \cdot 10 = 160$ códigos de área no formato *NYX* e $8 \cdot 10 \cdot 10 = 800$ códigos de área no formato *NXX*. Do mesmo modo, pela regra do produto, há $8 \cdot 8 \cdot 10 = 640$ códigos de função no formato *NNX*. A regra do produto também mostra que há $10 \cdot 10 \cdot 10 \cdot 10 = 10\,000$ códigos de localização no formato *XXXX*.

Conseqüentemente, aplicando a regra novamente, segue que no prefixo antigo há

$$160 \cdot 640 \cdot 10\,000 = 1\,024\,000\,000$$

números diferentes disponíveis na América do Norte. No prefixo novo há

$$800 \cdot 800 \cdot 10\,000 = 6\,400\,000\,000$$

números diferentes disponíveis. ◀

EXEMPLO 9 Qual o valor de k depois de o seguinte código ser executado?

```
k := 0
for i_1 := 1 to n_1
    for i_2 := 1 to n_2
        .
        .
        .
      for i_m := 1 to n_m
          k := k + 1
```

Solução: O valor inicial de k é zero. Cada vez que um laço é dado, 1 é adicionado a k. Considere T_i como tarefa de dar o i-ésimo laço. Então, o número de vezes que os laços são dados é o número de formas necessárias para realizar as tarefas $T_1, T_2, \ldots, T_m$. O número de formas para executar a tarefa T_j, $j = 1, 2, \ldots, m$, é n_j, porque j-ésimo laço é dado, uma vez para cada inteiro i_j, com $1 \leq i_j \leq n_j$. Pela regra do produto, segue que o laço é dado $n_1, n_2 \ldots n_m$ vezes. Assim, o valor final de k é $n_1 n_2 \ldots n_m$. ◀

EXEMPLO 10 **Contagem de Subconjuntos de um Conjunto Finito** Use a regra do produto para mostrar que o número de subconjuntos diferentes de um conjunto finito S é $2^{|S|}$.

Solução: Considere S como um conjunto finito. Liste os elementos de S em ordem arbitrária. Retome a Seção 2.2, que contém a correspondência um para um entre subconjuntos de S, que é uma cadeia de bits de extensão $|S|$. Isto é, um subconjunto de S associa-se com a cadeia de bits com um 1 na posição i se o i-ésimo elemento na lista pertence ao subconjunto e um 0, caso contrário. Pela regra do produto, há $2^{|S|}$ cadeias de bits de extensão $|S|$. Assim, $|P(S)| = 2^{|S|}$. ◀

A regra do produto é freqüentemente usada com conjuntos desta forma: Se $A_1, A_2, \ldots, A_m$ são conjuntos finitos, então o número de elementos no produto cartesiano desses conjuntos é o produto do número de elementos em cada conjunto. Relacionando isto à regra do produto, note que a tarefa de escolher um elemento no produto cartesiano $A_1 \times A_2 \times \ldots \times A_m$ é feita escolhendo-se um elemento em A_1, um elemento em $A_2, \ldots$, e um elemento em A_m. Pela regra do produto, segue que

$$|A_1 \times A_2 \times \ldots \times A_m| = |A_1| \cdot |A_2| \ldots |A_m|.$$

Introduziremos agora a regra da soma.

REGRA DA SOMA Se uma tarefa puder ser realizada em uma das n_1 formas ou em uma das n_2 formas, em que nenhum dos elementos do conjunto das n_1 formas é o mesmo que algum elemento do conjunto das n_2 formas, então há $n_1 + n_2$ formas de realizar a tarefa.

O Exemplo 11 ilustra como a regra da soma é usada.

EXEMPLO 11 Suponha que um membro da faculdade de matemática ou um estudante que tenha mestrado em matemática seja escolhido como representante para um comitê da Universidade. Quantas escolhas diferentes podem ser feitas para esse representante se houver 37 membros da faculdade de matemática e 83 mestres em matemática e nenhum for ao mesmo tempo um membro da faculdade e um mestre?

Solução: Há 37 maneiras de escolher um membro da faculdade de matemática e há 83 maneiras de escolher um estudante que é mestre em matemática. Escolher um membro da faculdade de

matemática não é o mesmo que escolher um estudante que é mestre em matemática porque nenhum deles é ao mesmo tempo um membro da faculdade e um mestre. Então, pela regra da soma, temos que há $37 + 83 = 120$ maneiras possíveis de escolher esse representante. ◀

Podemos estender a regra da soma a mais de duas tarefas. Suponha que uma tarefa possa ser realizada de n_1 maneiras, de n_2 maneiras, . . . , ou de n_m maneiras, em que nenhuma das formas do grupo n_i de realizar a tarefa é a mesma da do grupo de n_j, para todos os pares i e j com $1 \leq i < j \leq m$. Então o número de maneiras de realizar a tarefa é $n_1 + n_2 + \cdots + n_m$. Esta versão estendida da regra da soma é sempre útil em problemas de contagem, como os exemplos 12 e 13 mostram. Esta versão da regra pode ser demonstrada usando-se a indução matemática a partir da regra da soma para dois conjuntos. (Isso é apresentado no Exercício 59 no final da seção.)

EXEMPLO 12 Um estudante pode escolher um projeto de computação a partir de três listas. As três listas contêm 23, 15 e 19 projetos possíveis, respectivamente. Nenhum projeto está em mais de uma lista. Quantos projetos possíveis existem para serem escolhidos?

Solução: O estudante pode escolher um projeto da primeira lista, um projeto da segunda lista ou um projeto da terceira lista. Como todos os projetos estão em apenas uma lista, pela regra da soma há $23 + 15 + 19 = 57$ maneiras de escolher um projeto. ◀

EXEMPLO 13 Qual o valor de k depois de o seguinte código ser executado?

```
k := 0
for i_1 := 1 to n_1
    k := k + 1
for i_2 := 1 to n_2
    k := k + 1
     .
     .
     .
for i_m := 1 to n_m
    k := k + 1
```

Solução: O valor inicial de k é zero. Este conjunto de códigos é composto de m laços diferentes. Cada vez que um laço é dado, 1 é adicionado a k. Para determinar o valor de k depois que esse código tiver sido executado, precisamos determinar quantas vezes são dados os laços. Note que há n_i maneiras de realizar o i-ésimo laço. Como é dado apenas um laço por vez, a regra da soma mostra que o valor final de k, que é o número de maneiras de relaizar m laços, é $n_1 + n_2 + \cdots + n_m$. ◀

A regra da soma pode ser escrita em termos de conjuntos por: Se $A_1, A_2, \ldots, A_m$ são conjuntos finitos disjuntos, então o número de elementos na união desses conjuntos é a soma dos números de elementos nos conjuntos. Para relacionar isto à nossa proposição da regra da soma, note que há $|A_i|$ maneiras para escolher um elemento de A_i para $i = 1, 2, \ldots, m$. Sendo os conjuntos disjuntos, quando selecionamos um elemento de um dos conjuntos A_i, não podemos selecionar também um elemento de um conjunto diferente A_j. Conseqüentemente, pela regra da soma, por não existir elementos comuns em dois desses conjuntos, o número de maneiras de escolher um elemento de um dos conjuntos, que é o número de elementos da união, é

$$|A_1 \cup A_2 \cup \cdots \cup A_m| = |A_1| + |A_2| + \cdots + |A_m|.$$

Esta igualdade apenas se aplica quando os conjuntos em questão são disjuntos. A situação é muito mais complicada quando esses conjuntos têm elementos em comum. Tal situação será brevemente discutida mais tarde nesta seção e mais profundamente no Capítulo 7.

Problemas Mais Complexos de Contagem

Muitos problemas de contagem não podem ser resolvidos usando apenas as regra de soma ou de produto. Entretanto, muitos problemas complicados de contagem podem ser resolvidos usando-se as duas regras combinadas.

EXEMPLO 14 Em uma versão da linguagem computacional BASIC, o nome de uma variável é uma seqüência de um ou dois caracteres alfanuméricos, em que letras maiúsculas e minúsculas não são distinguidas. (Um caractere *alfanumérico* é aquele que utiliza as 26 letras do alfabeto da língua inglesa ou um dos 10 dígitos.) Além disso, um nome variável deve começar com uma letra e deve ser diferente das cinco seqüências de dois caracteres que são reservadas para o uso em programação (os comandos). Quantos nomes diferentes de variáveis são possíveis nesta versão do BASIC?

Exemplos Extras

Solução: Considere V como o número de nomes de variáveis diferentes nesta versão do BASIC. Considere V_1 como o número de nomes com um caractere e V_2 como o número de nomes com dois caracteres. Então, pela regra da soma, $V = V_1 + V_2$. Note que $V_1 = 26$ porque um caractere único deve ter uma letra. Mais que isso, pela regra do produto, há $26 \cdot 36$ seqüências de extensão dois que começam com um letra e terminam com um caractere alfanumérico. Entretanto, cinco dessas seqüências devem ser excluídas, então $V_2 = 26 \cdot 36 - 5 = 931$. Assim, há $V = V_1 + V_2 = 26 + 931 = 957$ nomes diferentes para variáveis nesta versão do BASIC. ◀

EXEMPLO 15 Cada usuário em um sistema computacional tem uma senha, com seis a oito caracteres, em que cada caractere é uma letra maiúscula ou um dígito numérico. Cada senha contém pelo menos um dígito numérico. Quantas senhas são possíveis?

Solução: Considere P como o número total de senhas possíveis e P_6, P_7 e P_8 como o número de senhas possíveis com 6, 7 e 8 caracteres, respectivamente. Pela regra da soma, $P = P_6 + P_7 + P_8$. Encontraremos agora P_6, P_7 e P_8. Encontrar P_6 diretamente é difícil. Para descobrir P_6 é mais fácil encontrar o número de seqüência de letras maiúsculas e dígitos que formam a combinação com seis caracteres, incluindo aqueles com nenhum dígito, e subtrair do número de seqüências sem dígitos. Pela regra do produto, o número de seqüências de seis caracteres é 36^6 e o número de seqüências sem dígitos é 26^6. Assim,

$$P_6 = 36^6 - 26^6 = 2\,176\,782\,336 - 308\,915\,776 = 1\,867\,866\,560.$$

Analogamente, podemos mostrar que

$$P_7 = 36^7 - 26^7 = 78\,364\,164\,096 - 8\,031\,810\,176 = 70\,332\,353\,920$$

e

$$P_8 = 36^8 - 26^8 = 2\,821\,109\,907\,456 - 208\,827\,064\,576 = 2\,612\,282\,842\,880.$$

Conseqüentemente,

$$P = P_6 + P_7 + P_8 = 2\,684\,483\,063\,360.$$ ◀

Número de Bits	0	1	2	3	4	8	16	24	31
Classe A	0	netid				hostid			
Classe B	1	0	netid				hostid		
Classe C	1	1	0	netid				hostid	
Classe D	1	1	1	0	Endereço Multicast				
Classe E	1	1	1	1	0	Endereço			

FIGURA 1 Endereços de Internet (IPv4).

EXEMPLO 16 **Contagem de Endereços da Internet** Na Internet, que é construída a partir de redes de transmissão de computadores interconectadas fisicamente, cada computador (ou, mais precisamente, cada rede de transmissão conectada a um computador) possui um *endereço de Internet*. Na Versão 4 do Protocolo de Internet (IPv4), agora em uso, um endereço é uma cadeia de 32 bits. Ele começa com um *número de transmissão* (*netid*). O netid é seguido de um *número de hospedagem* (*hostid*), que identifica um computador como um membro de uma determinada rede de transmissão.

Links

Três formas de endereços são usadas, com números diferentes de bits usados por netids e hostids. **Endereços Classe A**, usados pelas maiores redes de transmissão, consistem em 0, seguido de um netid de 7 bits e um hostid de 24 bits. **Endereços Classe B**, usados por redes medianas, consistem em 10, seguido por um netid de 14 bits e um hostid de 16 bits. **Endereços Classe C**, usados pelas redes menores, consistem em 110, seguido por um netid de 21 bits e um hostid de 8 bits. Há muitas restrições nos endereços para os usos especiais: 1111111 não está disponível no netid de uma Classe A e os hostids de 0 e 1 não estão disponíveis para o uso em nenhuma rede. Um computador na Internet tem ou um endereço Classe A, ou um Classe B, ou um Classe C. (Além dos endereços de Classe A, B e C, há também os endereços de Classe D, reservados para o uso em transmissão múltipla, quando vários computadores são endereçados ao mesmo tempo, que consistem em 1110 seguido de 28 bits, e os endereços de Classe E, reservados para uso futuro, que consistem em 11110 seguido de 27 bits. Nem os endereços de Classe D nem os de Classe E são considerados endereços de IP de um computador na Internet.) A Figura 1 mostra os endereços IPv4. (Limitações no número de netids da Classe A e da Classe B tornaram os endereços IPv4 inadequados; IPv6, uma nova versão de IP, usa endereços de 128 bits para resolver este problema.)

Quantos endereços IPv4 diferentes estão disponíveis para os computadores na Internet?

Solução: Considere x como o número de endereços disponíveis para computadores na Internet e x_A, x_B e x_C como o número de endereços da Classe A, Classe B e Classe C disponíveis, respectivamente. Pela regra da soma, $x = x_A + x_B + x_C$.

Para encontrar x_A, note que há $2^7 - 1 = 127$ netids de Classe A, relembrando que o netid 1111111 não está disponível. Para cada netid, há $2^{24} - 2 = 16\,777\,214$ hostids, relembrando que todos os hostids que tenham somente 0s e 1s não estão disponíveis. Conseqüentemente, $x_A = 127 \cdot 16\,777\,214 = 2\,130\,706\,178$.

Para encontrar x_B e x_C, note que há $2^{14} = 16\,384$ netids de Classe B e $2^{21} = 2\,097\,152$ netids de Classe C. Para cada netid de Classe B, há $2^{16} - 2 = 65\,534$ hostids, e para cada netid de Classe C, há $2^8 - 2 = 254$ hostids, relembrando que em cada rede de transmissão os hostids compostos somente de 0s e 1s não estão disponíveis. Conseqüentemente, $x_B = 1\,073\,709\,056$ e $x_C = 532\,676\,608$.

Concluímos que o número total de endereços IPv4 disponíveis é $x = x_A + x_B + x_C = 2\,130\,706\,178 + 1\,073\,709\,056 + 532\,676\,608 = 3\,737\,091\,842$. ◀

Princípio da Inclusão–Exclusão

Suponha que uma tarefa possa ser realizada de n_1 ou n_2 maneiras, mas que algumas das maneiras do conjunto n_1 para realizar as tarefas são as mesmas de algumas das do n_2. Neste caso, não podemos usar a regra da soma para contar o número de maneiras de realizar a tarefa. Adicionar o número de

maneiras de realizar as tarefas nessas duas formas leva a uma contagem excessiva, porque as maneiras de realizar a tarefa naquelas formas que são idênticas serão contadas duas vezes. Para contar corretamente o número de maneiras para realizar as duas tarefas, somamos o número total de maneiras para realizá-la, e então subtraímos o número de maneiras de realizar a tarefa que pertence aos dois grupos. Esta técnica é chamada de **princípio da inclusão–exclusão**. Em alguns casos, este princípio também pode ser chamado de **princípio da subtração** para contagem. O Exemplo 17 ilustra como podemos resolver os problemas de contagem usando este princípio.

EXEMPLO 17 Quantas cadeias de bits de extensão oito começam com 1 bit ou terminam com dois bits 00?

Exemplos Extras

Solução: Podemos construir uma cadeia de bits de extensão oito que ou começa com um 1 bit ou termina com dois bits 00 construindo uma cadeia de bits de extensão oito que começa com um 1 bit ou construindo uma cadeia de bits de extensão oito que termina com dois bits 00. Podemos construir uma cadeia de bits que começa com 1, de $2^7 = 128$ maneiras, seguindo a regra do produto, pois o primeiro bit pode ser escolhido de uma única maneira e cada um dos outros sete bits podem ser escolhidos de duas formas. Analogamente, podemos construir uma cadeia de bits de extensão oito que termina com dois bits 00, de $2^6 = 64$ maneiras. Seguimos a regra do produto, pois cada um dos primeiros seis bits pode ser escolhido de duas maneiras e os últimos dois bits podem ser escolhidos apenas de uma maneira.

1 _ _ _ _ _ _ _ ($2^7 = 128$ maneiras)

_ _ _ _ _ _ 0 0 ($2^6 = 64$ maneiras)

1 _ _ _ _ _ 0 0 ($2^5 = 32$ maneiras)

Algumas maneiras de construir uma cadeia de bits de extensão oito que começa com 1 são as mesmas de construir uma cadeia de bits de extensão oito que termina com dois bits 00. Há $2^5 = 32$ maneiras de construir tal cadeia seguindo a regra do produto, porque o primeiro bit pode ser escolhido apenas de uma forma, e do segundo até o sexto bit podem ser escolhidos de duas maneiras e os dois últimos bits podem ser escolhidos de uma única forma. Conseqüentemente, o número de cadeias de bits de extensão oito que começa com 1 ou termina com 00, que é igual ao número de maneiras de construir uma cadeia de bits de extensão oito que começa com 1 ou termina com 00, é $128 + 64 - 32 = 160$. ◀

Podemos descrever este princípio de contagem em termos de conjuntos. Considere A_1 e A_2 como conjuntos. Há $|A_1|$ maneiras de selecionar um elemento do conjunto A_1 e $|A_2|$ maneiras de selecionar um elemento do conjunto A_2. O número de maneiras de escolher um elemento do conjunto A_1 ou do conjunto A_2, que é o número de maneiras de escolher um elemento da união desses conjuntos, é a soma do número de maneiras de selecionar um elemento de A_1 com o número de maneiras de selecionar um elemento de A_2, menos o número de maneiras de selecionar um elemento que pertence aos dois conjuntos, A_1 e A_2. Visto que existem $|A_1 \cup A_2|$ maneiras de escolher um elemento ou em A_1 ou em A_2 e $|A_1 \cap A_2|$ maneiras de selecionar um elemento comum dos dois conjuntos, temos que

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$

Esta é a fórmula dada na Seção 2.2 para o número de elementos na união de dois conjuntos.

Apresentaremos um exemplo que ilustra como a formulação do princípio de inclusão–exclusão pode ser usado para resolver problemas de contagem.

EXEMPLO 18 Uma companhia de computadores recebe 350 currículos de graduados em computação para uma proposta de trabalho de planejamento de uma nova linha de servidores da Web. Suponha que 220 dessas pessoas sejam da área de ciência da computação, 147 de administração e 51 de ciência da computação e administração. Quantos desses graduados não são da área de ciência da computação nem de administração?

Solução: Para encontrar o número de candidatos que não são graduados nem em ciência da computação nem em administração, podemos subtrair o número de estudantes que são graduados em ciência da computação ou em administração (ou ambos) do total de candidatos. Considere A_1 como o conjunto de estudantes graduados em ciência da computação e A_2, o conjunto dos estudantes graduados em administração. Então, $A_1 \cup A_2$ é o conjunto dos graduandos em ciência da computação ou em administração (ou ambos) e $A_1 \cap A_2$ é o conjunto de estudantes graduados em

ciência da computação e em administração. Pelo princípio da inclusão–exclusão, o número de estudantes graduados em ciência da computação ou em administração (ou ambos) é igual a

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2| = 220 + 147 - 51 = 316.$$

Concluímos que $350 - 316 = 34$ dos currículos recebidos não são de graduados nem em ciência da computação nem em administração. ◀

O princípio da inclusão–exclusão pode ser generalizado para encontrar o número de maneiras de realizar n tarefas diferentes ou, de modo equivalente, encontrar o número de elementos da união de n conjuntos, sempre que n for um número inteiro e positivo. Estudaremos o princípio da inclusão–exclusão e algumas de suas aplicações no Capítulo 7.

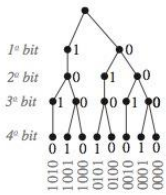


FIGURA 2
Cadeia de Bits de Comprimento Quatro sem 1s Consecutivos.

Diagramas de Árvore

Problemas de contagem podem ser resolvidos usando-se os **diagramas de árvore.** Uma árvore consiste em uma raiz, um número de galhos que partem da raiz e possíveis galhos adicionais no final de cada galho. (Estudaremos as árvores com detalhes no Capítulo 10.) Para usar as árvores na contagem, usamos um galho para representar cada escolha possível. Representamos as possíveis saídas pelas folhas, que são os finais dos galhos sem outros que comecem a partir deles.

Note que um diagrama de árvore é usado para resolver um problema de contagem, sendo que o número de escolhas necessárias para alcançar uma folha pode variar (veja o Exemplo 20, por exemplo).

EXEMPLO 19 Quantas cadeias de bits de comprimento quatro não possuem dois números 1s consecutivos?

Solução: O diagrama de árvore da Figura 2 mostra todas as cadeias de bits de comprimento quatro sem dois números 1s consecutivos. Vemos que há oito cadeias de bits de comprimento quatro sem dois números 1s consecutivos. ◀

EXEMPLO 20 Uma eliminatória entre dois times é composta de no máximo cinco jogos. O primeiro time que vence três jogos, vence a eliminatória. De quantas formas diferentes a eliminatória pode ocorrer?

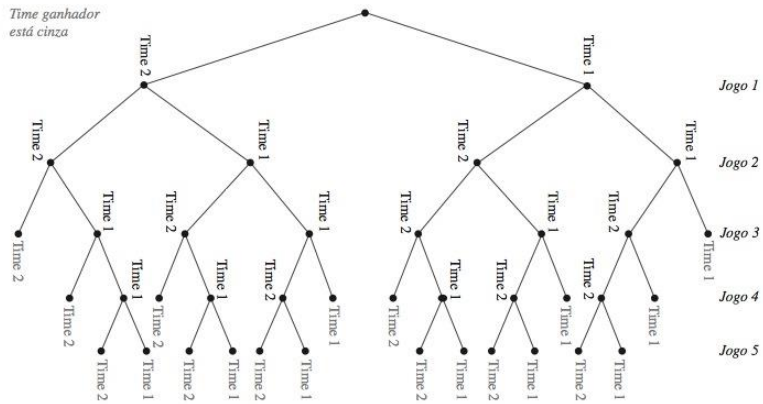


FIGURA 3 **Os três melhores jogos de uma eliminatória com cinco partidas.**

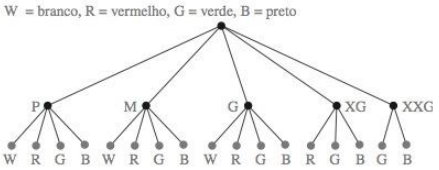


FIGURA 4 Contando Variedades de Camisetas.

Solução: O diagrama de árvore da Figura 3 mostra todas as formas de ocorrer a eliminatória, apresentando o vencedor de cada partida. Vemos que há 20 maneiras diferentes de a eliminatória acontecer. ◀

EXEMPLO 21 Suponha que as camisetas "Eu Amo Florianópolis" tenham cinco tamanhos diferentes: P, M, G, XG e XXG. Além disso, cada tamanho vem em quatro cores: branco, vermelho, verde e preto, exceto para o tamanho XG, que vem apenas em vermelho, verde e preto, e o tamanho XXG, que vem apenas em verde e preto. Quantas camisetas diferentes uma loja terá em estoque para ter ao menos uma camiseta de cada cor e tamanho disponível?

Solução: O diagrama de árvore da Figura 4 mostra todos os tamanhos e cores disponíveis. Partindo dele, o dono da loja deve ter em estoque de 17 camisetas diferentes. ◀

Exercícios

1. Em uma universidade, há 18 graduados em matemática e 325 em ciência da computação.
   a) Há quantas maneiras de escolher dois representantes de modo que um seja matemático e o outro um cientista da computação?
   b) Há quantas maneiras de escolher um representante que seja ou matemático ou cientista da computação?
2. Um edifício empresarial contém 27 andares, com 37 escritórios em cada andar. Quantos escritórios há no prédio?
3. Um exame de múltipla escolha contém 10 questões. Há quatro respostas possíveis para cada questão.
   a) De quantas maneiras um estudante pode responder às questões do exame se responder a todas as questões?
   b) De quantas maneiras um estudante pode responder às questões do exame se ele pode deixar respostas em branco?
4. Uma determinada marca de camisetas vem em 12 cores, tem modelos masculino e feminino e três tamanhos para cada sexo. Quantos tipos diferentes de camisetas são fabricados?
5. Seis companhias aéreas voam de Nova York a Denver e sete de Denver a São Francisco. Quantos pares diferentes de companhias aéreas você pode escolher para fazer uma viagem de Nova York a São Francisco, com conexão em Denver, em que você escolhe uma companhia para viajar até Denver e outra para continuar o vôo a São Francisco? Quantas dessas maneiras envolvem mais de uma companhia?
6. Há quatro auto-estradas principais de Boston a Detroit e seis de Detroit a Los Angeles. Quantas auto-estradas há para o percurso Boston–Los Angeles, via Detroit?
7. Quantas iniciais diferentes com três letras uma pessoa pode ter?
8. Quantas iniciais diferentes com três letras sem que nenhuma seja repetida uma pessoa pode ter?
9. Quantas iniciais diferentes com três letras que comecem com a letra A são possíveis?
10. Quantas cadeias de bits de comprimento oito são possíveis?
11. Quantas cadeias de bits de comprimento 10 que comecem e terminem com um 1 são possíveis?
12. Quantas cadeias de bits de comprimento seis ou menos são possíveis?
13. Quantas cadeias de bits com comprimento não excedente a n, em que n é um número inteiro positivo, compostas apenas de 1s são possíveis?
14. Quantas cadeias de bits com comprimento n, em que n é um número inteiro positivo, que começam e terminam com 1 são possíveis?
15. Quantas seqüências de letras minúsculas de extensão quatro ou menos são possíveis?
16. Quantas seqüências de quatro letras minúsculas que contenham a letra x são possíveis?
17. Quantas seqüências de cinco caracteres ASCII contêm o caractere @ ("arroba") ao menos uma vez? (*Nota:* Há 128 caracteres ASCII diferentes.)
18. Quantos números inteiros positivos entre 5 e 31
    a) são divisíveis por 3? Quais números inteiros são esses?
    b) são divisíveis por 4? Quais números inteiros são esses?
    c) são divisíveis por 3 e por 4? Quais números inteiros são esses?

19. Quantos números inteiros positivos entre 50 e 100
- **a)** são divisíveis por 7? Quais números inteiros são esses?
- **b)** são divisíveis por 11? Quais números inteiros são esses?
- **c)** são divisíveis por 7 e por 11? Quais números inteiros são esses?

20. Quantos números inteiros positivos menores que 1000
- **a)** são divisíveis por 7?
- **b)** são divisíveis por 7, mas não por 11?
- **c)** são divisíveis por 7 e por 11?
- **d)** são divisíveis por 7 ou por 11?
- **e)** são divisíveis exatamente por 7 e por 11?
- **f)** não são divisíveis nem por 7 nem por 11?
- **g)** têm dígitos distintos?
- **h)** têm dígitos distintos e são pares?

21. Quantos números inteiros positivos entre 100 e 999, incluindo este último,
- **a)** são divisíveis por 7?
- **b)** são ímpares?
- **c)** têm os três dígitos iguais?
- **d)** não são divisíveis por 4?
- **e)** são divisíveis por 3 ou por 4?
- **f)** não são divisíveis por 3 nem por 4?
- **g)** são divisíveis por 3, mas não por 4?
- **h)** são divisíveis por 3 e por 4?

22. Quantos números inteiros positivos entre 1000 e 9999, incluindo este último,
- **a)** são divisíveis por 9?
- **b)** são pares?
- **c)** têm dígitos distintos?
- **d)** não são divisíveis por 3?
- **e)** são divisíveis por 5 ou por 7?
- **f)** não são divisíveis por 5 nem por 7?
- **g)** são divisíveis por 5, mas não por 7?
- **h)** são divisíveis por 5 e por 7?

23. Quantas seqüências de três dígitos decimais
- **a)** não são formadas pelo mesmo dígito três vezes?
- **b)** começam com um dígito ímpar?
- **c)** têm exatamente dois dígitos que são o número 4?

24. Quantas seqüências de quatro dígitos decimais
- **a)** não contêm o mesmo dígito duas vezes?
- **b)** terminam com um dígito par?
- **c)** têm exatamente três dígitos que são o número 9?

25. Um comitê é formado por um representante de cada um dos 50 estados dos Estados Unidos, no qual o representante do estado é ou o governador ou um dos dois senadores daquele determinado estado. Quantas maneiras há para formar esse comitê?

26. Quantas placas de identificação de veículos podem ser feitas usando-se três dígitos seguidos de três letras ou três letras seguidas de três dígitos?

27. Quantas placas de identificação de veículos podem ser feitas usando-se duas letras seguidas de quatro dígitos ou dois dígitos seguidos de quatro letras?

28. Quantas placas de identificação de veículos podem ser feitas usando-se três letras seguidas de três dígitos ou quatro letras seguidas de dois dígitos?

29. Quantas placas de identificação de veículos podem ser feitas usando-se duas ou três letras seguidas de dois ou três dígitos?

30. Quantas seqüências de oito letras do alfabeto da língua inglesa são possíveis
- **a)** se as letras puderem ser repetidas?
- **b)** se nenhuma letra puder ser repetida?
- **c)** que comecem com X, se as letras puderem ser repetidas?
- **d)** que comecem com X, se nenhuma letra puder ser repetida?
- **e)** que comecem e terminem com X, se as letras puderem ser repetidas?
- **f)** que comecem com as letras BO (nesta ordem), se as letras puderem ser repetidas?
- **g)** que comecem e terminem com as letras BO (nesta ordem), se as letras puderem ser repetidas?
- **h)** que comecem ou terminem com as letras BO (nesta ordem), se as letras puderem ser repetidas?

31. Quantas seqüências de oito letras do alfabeto da língua inglesa são possíveis
- **a)** que não contenham vogais, se as letras puderem ser repetidas?
- **b)** que não contenham vogais, se as letras não puderem ser repetidas?
- **c)** que comecem com uma vogal, se as letras puderem ser repetidas?
- **d)** que comecem com uma vogal, se as letras não puderem ser repetidas?
- **e)** que contenham pelo menos uma vogal, se as letras puderem ser repetidas?
- **f)** que contenham exatamente uma vogal, se as letras puderem ser repetidas?
- **g)** que comecem com X e que contenham pelo menos uma vogal, se as letras puderem ser repetidas?
- **h)** que comecem e terminem com X e contenham pelo menos uma vogal, se as letras puderem ser repetidas?

32. Quantas funções diferentes são possíveis a partir de um conjunto de 10 elementos para os conjuntos com os seguintes números de elementos?
a) 2 **b)** 3 **c)** 4 **d)** 5

33. Quantas funções injetoras são possíveis a partir de um conjunto com cinco elementos para os conjuntos com os seguintes números de elementos?
a) 4 **b)** 5 **c)** 6 **d)** 7

34. Quantas funções são possíveis a partir do conjunto $\{1, 2, \ldots, n\}$, em que n é um número inteiro e positivo, para o conjunto $\{0, 1\}$?

35. Quantas funções são possíveis a partir do conjunto $\{1, 2, \ldots, n\}$, em que n é um número inteiro e positivo, para o conjunto $\{0, 1\}$
- **a)** que são injetoras?
- **b)** que atribuam 0 aos valores 1 e n?
- **c)** que atribuam 1 a exatamente um dos números inteiros positivos menores que n?

36. Quantas funções parciais (veja o preâmbulo do Exercício 73 na Seção 2.3) são possíveis a partir de um conjunto com cinco elementos para um conjunto que tenha cada um dos números de elementos abaixo?
a) 1 **b)** 2 **c)** 5 **d)** 9

37. Quantas funções parciais (veja o preâmbulo do Exercício 73 na Seção 2.3) são possíveis a partir de um conjunto com m elementos para um conjunto com n elementos, em que m e n são números inteiros positivos?

38. Quantos subconjuntos de um conjunto com 100 elementos têm mais de um elemento?

39. Um **palíndromo** é uma seqüência no qual o seu inverso é idêntico à seqüência. Quantas cadeias de bits de extensão n são palíndromos?

40. De quantas maneiras um fotógrafo pode organizar 6 pessoas de um grupo de 10, em um casamento, em que a noiva e o noivo estão entre estas 10 pessoas, se

a) a noiva deve estar na foto?

b) tanto a noiva quanto o noivo devem estar na foto?

c) um dos dois, noiva ou noivo, deve estar na foto?

41. De quantas maneiras um fotógrafo pode organizar seis pessoas em uma fila, em um casamento, incluindo a noiva e o noivo, se

a) a noiva deve estar ao lado do noivo?

b) a noiva não está ao lado do noivo?

c) a noiva está posicionada à esquerda do noivo?

42. Quantas cadeias de bits de comprimento sete começam com dois 0s ou terminam com três 1s?

43. Quantas cadeias de bits de comprimento 10 começam com três 0s ou terminam com dois 0s?

***44.** Quantas cadeias de bits de comprimento 10 têm cinco 0s consecutivos ou cinco 1s consecutivos?

****45.** Quantas cadeias de bits de comprimento oito têm três 0s consecutivos ou quatro 1s consecutivos?

46. Todo estudante em uma aula de matemática discreta é ou da área de ciência da computação ou de matemática ou é das duas áreas. Quantos estudantes há na sala se 38 são graduados em ciência da computação (incluindo os que são das duas áreas), 23 graduados em matemática (incluindo os das duas áreas) e 7 graduados nas duas áreas?

47. Quantos números inteiros positivos, menores ou iguais a 100, são divisíveis por 4 ou por 6?

48. Quantas iniciais diferentes uma pessoa pode ter se ela tiver pelo menos duas, mas no máximo cinco, iniciais diferentes? Suponha que cada inicial é uma das 26 letras do alfabeto da língua inglesa.

49. Suponha que uma senha para um sistema computacional deva ter pelo menos 8, mas não mais de 12 caracteres, em que cada caractere na senha é uma letra minúscula, uma letra maiúscula, um dígito ou um dos seis caracteres especiais *, >, <, !, +, e =.

a) Quantas senhas diferentes estão disponíveis para esse sistema?

b) Quantas dessas senhas contêm pelo menos uma ocorrência de pelo menos um dos seis caracteres especiais?

c) Se demora um nanossegundo para um hacker checar se cada senha possível é a sua senha, quanto tempo demoraria para o hacker tentar todas as senhas possíveis?

50. O nome de uma variável na linguagem de programação C é uma seqüência que pode conter letras maiúsculas, minúsculas, dígitos ou negrito. Além disso, o primeiro caractere na seqüência deve ser uma letra, maiúscula ou minúscula, ou um negrito. Se o nome da variável é determinado pelos seus primeiros oito caracteres, quantas variáveis diferentes podem ser nomeadas em C? (Note que o nome da variável pode conter menos que oito caracteres.)

51. Suponha que em um tempo futuro, a todo telefone no mundo seja atribuído um número que contenha um código do país de 1 a 3 dígitos, ou seja, sob a forma *X*, *XX* ou *XXX*, seguido do número de telefone com 10 dígitos, na forma de *NXX-NXX-XXXX* (como descrito no Exemplo 8). Quantos números de telefone diferentes estariam disponíveis na rede mundial, de acordo com este plano numérico?

52. Use um diagrama de árvore para encontrar o número de cadeias de bits de comprimento quatro sem três zeros consecutivos.

53. Quantas maneiras de organizar as letras a, b, c e d, tal que a não seja seguida imediatamente por b, são possíveis?

54. Use um diagrama de árvore para encontrar o número de maneiras para uma final de campeonato acontecer, em que o primeiro time que vencer quatro jogos de sete, vence o campeonato.

55. Use um diagrama de árvore para determinar o número de subconjuntos de {3, 7, 9, 11, 24} com a propriedade de que a soma dos elementos no subconjunto seja menor que 28.

56. a) Suponha que uma loja venda seis variedades de refrigerantes: coca, guaraná, laranja, sucos, limonada e citrus. Use um diagrama de árvore para determinar o número de tipos diferentes de garrafas que a loja deve estocar para ter disponíveis todas as variedades em todos os tamanhos de garrafas, considerando-se que todas as variedades estão disponíveis em garrafas de 300 ml, apenas a limonada está disponível em garrafa de 500 ml e apenas a coca e o guaraná em garrafas de 600 ml e todos, menos a limonada e o citrus, estão disponíveis em garrafas de 1l?

b) Responda à questão do item (a) usando as regras de contagem.

57. a) Suponha que um modelo popular de tênis de corrida esteja disponível tanto para homens quanto para mulheres. O tênis feminino vem nos tamanhos 6, 7, 8 e 9, e o masculino, nos tamanhos 8, 9, 10, 11 e 12. O modelo masculino vem nas cores branco e preto, e o feminino, em branco, vermelho e preto. Use um diagrama de árvore para determinar o número de diferentes tênis que uma loja tem de estocar para ter, pelo menos, um par disponível de todas as cores e tamanhos para mulheres e para homens.

b) Responda à questão do item (a) usando as regras de contagem.

***58.** Use a regra do produto para mostrar que há 2^{2^n} tabelas-verdade diferentes para as proposições em n variáveis.

59. Use a indução matemática para demonstrar a regra da soma para m tarefas a partir da regra da soma para duas tarefas.

60. Use a indução matemática para demonstrar a regra do produto para m tarefas a partir da regra do produto para duas tarefas.

61. Quantas diagonais tem um polígono convexo com n lados? (Lembre-se de que um polígono é convexo se todos os segmentos de reta que unirem dois pontos no interior ou na margem do polígono estiverem inteiramente dentro deste polígono e que uma diagonal de um polígono é um segmento de reta que liga dois vértices que não são adjacentes.)

62. Os dados são transmitidos pela Internet em **datagramas**, que são estruturados em blocos de bits. Cada datagrama contém informação de cabeçalho organizada em um máximo

de 14 campos diferentes (especificando muitas coisas, incluindo a fonte e os endereços de destinatário) e a área de dados que contém os dados atuais que são transmitidos. Um dos 14 campos de cabeçalho é o **campo de cabeçalho estendido** (indicado por HLEN), que é caracterizado pelo protocolo por ter 4 bits e por especificar a extensão do cabeçalho em blocos de 32 bits. Por exemplo, se HLEN = 0110, o cabeçalho é composto por seis blocos de 32 bits. Outro componente dos 14 campos de cabeçalho é o **campo de extensão total** de 16 bits (indicado por EXTENSÃO TOTAL), que especifica a extensão em bits de todo o datagrama, incluindo os campos de cabeçalho e a área de dados. A extensão da área de dados é a extensão total do datagrama menos a extensão do cabeçalho.

a) O maior valor possível para a EXTENSÃO TOTAL (que é de 16 bits) determina a extensão máxima total em octetos (blocos de 8 bits) de um datagrama da Internet. Qual é esse valor?

b) O maior valor possível para o HLEN (que é de 4 bits) determina a extensão máxima total de cabeçalho em blocos de 32 bits. Qual é esse valor? Qual é a extensão máxima total de cabeçalho em octetos?

c) A extensão mínima (e mais comum) de cabeçalho é 20 octetos. Qual é a extensão máxima total em octetos da área de dados de um datagrama da Internet?

d) Quantas seqüências diferentes de octetos na área de dados podem ser transmitidas se a extensão do cabeçalho for 20 octetos e a extensão total for a máxima possível?

5.2 O Princípio da Casa dos Pombos

Introdução

Links

Suponha que um grupo de 20 pombos voe para dentro de 19 casas para empoleirarem-se. Visto que há 20 pombos e 19 casas, pelo menos uma dessas 19 casas deverão ter no mínimo dois pombos. Para ver por que isso é verdadeiro, note que se cada casa tiver no máximo um pombo, no máximo 19 pombos, um por casa, poderão ser acomodados. Isso ilustra um princípio geral, chamado de **princípio da casa dos pombos**, que afirma que se há mais pombos que casas, então deverá ter pelo menos uma casa com pelo menos dois pombos (veja a Figura 1). É claro que esse princípio se aplica a outros objetos além de pombos e casas.

TEOREMA 1 **O PRINCÍPIO DA CASA DOS POMBOS** Se k for um número inteiro positivo e $k + 1$ ou mais objetos são colocados dentro de k caixas, então há uma caixa que terá dois ou mais objetos.

Demonstração: Vamos demonstrar o princípio da casa dos pombos usando uma demonstração por contraposição. Suponha que nenhuma das k caixas contenha mais de um objeto. Então, o número total de objetos seria no máximo k. Esta é uma contradição, porque há pelo menos $k + 1$ objetos. ◁

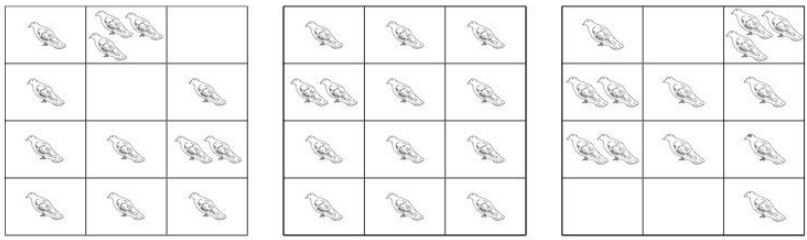


FIGURA 1 **Há mais Pombos que Casas.**

O princípio da casa dos pombos é também chamado de **Princípio das Gavetas de Dirichlet**, depois que o matemático alemão Dirichlet usou freqüentemente este princípio em seu trabalho, no século XIX.

O princípio da casa dos pombos pode ser usado para demonstrar diversas funções.

COROLÁRIO 1 Uma função f de um conjunto com $k + 1$ ou mais elementos para um conjunto com k elementos não é injetora.

Demonstração: Suponha que para cada elemento y no contradomínio de f temos uma caixa que contém todos os elementos x do domínio de f, tal que $f(x) = y$. Como o domínio contém $k + 1$ ou mais elementos e o contradomínio contém apenas k elementos, o princípio da casa dos pombos nos diz que uma das caixas contém dois ou mais elementos x do domínio. Isto significa que f não pode ser uma função injetora. ◁

Os exemplos 1 a 3 mostram como o princípio da casa dos pombos é usado.

EXEMPLO 1 Entre qualquer grupo de 367 pessoas, deverá existir ao menos duas com a mesma data de aniversário, porque há apenas 366 possíveis datas. ◀

EXEMPLO 2 Em qualquer grupo de 27 palavras em inglês, deverá haver pelo menos duas que começam com a mesma letra, pois há 26 letras no alfabeto da língua inglesa. ◀

EXEMPLO 3 Quantos estudantes deve haver em uma classe para garantir que pelo menos dois estudantes tenham a mesma nota nas provas finais, se a nota é graduada em uma escala de 0 a 100?

Solução: Há 101 notas possíveis no final. O princípio da casa dos pombos mostra que entre quaisquer 102 estudantes há pelo menos dois com a mesma nota. ◀

O princípio da casa dos pombos é um instrumento útil em muitas demonstrações, incluindo as demonstrações de resultados surpreendentes, como a do Exemplo 4.

EXEMPLO 4 Mostre que para todo número inteiro n há um múltiplo de n que contém apenas 0s e 1s quando escrito na base decimal.

Exemplos Extras

Solução: Considere n como um número inteiro positivo. Considere $n + 1$ números inteiros 1, 11, 111, . . . , 11 . . . 1 (em que o último número inteiro da lista é um número inteiro com $n + 1$ 1s em sua expansão decimal). Note que há n restos possíveis quando um número inteiro é dividido por n. Como, há $n + 1$ números inteiros nesta lista. Pelo princípio da casa dos pombos, deverá haver dois com o mesmo resto quando for dividido por n. O maior desses números inteiros menos o menor deles é um múltiplo de n, que tem um expansão decimal de 0s e 1s. ◀

Links

G. LEJEUNE DIRICHLET (1805–1859) G. Lejeune Dirichlet nasceu em uma família francesa que morava perto de Colônia, na Alemanha. Estudou na Universidade de Paris e ocupou cargos na Universidade de Breslau e na Universidade de Berlim. Em 1855, ele foi escolhido para ser o sucessor de Gauss na Universidade de Göttingen. Dirichlet é considerado a primeira pessoa a estudar a fundo a obra *Disquisitiones Arithmeticae*, de Gauss, publicada 20 anos antes. Ele mantinha uma cópia desse trabalho ao seu lado mesmo quando viajava. Dirichlet fez importantes descobertas sobre a teoria dos números, inclusive o teorema de que há números primos infinitos em progressões aritméticas $an + b$, quando a e b são primos entre si. Ele demonstrou o caso de $n = 5$ do último teorema de Fermat, afirmando que não há soluções não triviais em números inteiros para $x^5 + y^5 = z^5$. Dirichlet fez também muitas contribuições em análise.

A Generalização do Princípio da Casa dos Pombos

O princípio da casa dos pombos afirma que deverá haver pelo menos dois objetos na mesma caixa quando houver mais objetos que caixas. No entanto, mais pode ser dito quando o número de objetos excede um múltiplo do número de caixas. Por exemplo, entre qualquer conjunto de 21 dígitos decimais, deverá haver 3 que serão os mesmos. Isto ocorre porque quando 21 objetos são distribuídos em 10 caixas, uma caixa deverá conter mais que 2 objetos.

TEOREMA 2 **A GENERALIZAÇÃO DO PRINCÍPIO DA CASA DOS POMBOS** Se N objetos são colocados em k caixas, então há pelo menos uma caixa com $\lceil N/k \rceil$ objetos.

Demonstração: Usaremos uma demonstração por contradição. Suponha que nenhuma das caixas tenha mais que $\lceil N/k \rceil - 1$ objetos. Então, o número total de objetos é no máximo

$$k\left(\left\lceil \frac{N}{k} \right\rceil - 1\right) < k\left(\left(\frac{N}{k} + 1\right) - 1\right) = N,$$

em que a inequação $\lceil N/k \rceil < (N/k) + 1$ foi usada. Isto é uma contradição, pois há um total de N objetos. ◁

Um tipo comum de problema pede o número mínimo de objetos, tal que pelo menos r desses objetos deverão estar em uma das k caixas quando esses objetos forem distribuídos entre as caixas. Quando temos N objetos, a generalização do princípio da casa dos pombos diz que deverá haver pelo menos r objetos em uma das caixas, contanto que $\lceil N/k \rceil \geq r$. O menor número inteiro N com $N/k > r - 1$, ou seja, $N = k(r - 1) + 1$, é o menor número inteiro que satisfaz a inequação $\lceil N/k \rceil \geq r$. Um menor valor de N poderia ser suficiente? A resposta é não, porque se temos $k(r - 1)$ objetos, poderíamos colocar $r - 1$ deles em cada uma das k caixas e nenhuma caixa teria pelo menos r objetos.

Quando pensamos em problemas desse tipo, é útil considerar como você pode evitar ter pelo menos r objetos em uma das caixas quando você coloca objetos sucessivos. Para evitar colocar um r-ésimo objeto em qualquer caixa, você deve eventualmente terminar com $r - 1$ objetos em cada caixa. Não há como adicionar o próximo objeto sem colocar um r-ésimo objeto naquela caixa.

Os exemplos 5 a 8 ilustram como a generalização do princípio da casa dos pombos é aplicada.

EXEMPLO 5 Entre 100 pessoas, há pelo menos $\lceil 100/12 \rceil = 9$ que nasceram no mesmo mês. ◀

EXEMPLO 6 Qual o menor número de estudantes necessário em uma sala de matemática discreta para se ter certeza de que pelo menos seis receberão a mesma nota, se são possíveis cinco notas, A, B, C, D e F?

Exemplos Extras

Solução: O número mínimo de estudantes necessário para garantir que pelo menos seis estudantes tirem a mesma nota é o menor número inteiro N, tal que $\lceil N/5 \rceil = 6$. Assim, o menor número inteiro é $N = 5 \cdot 5 + 1 = 26$. Se você tiver apenas 25 estudantes, é possível que haja cinco que receberão cada nota, e não seis. Assim, 26 é o menor número de estudantes necessário para assegurar que pelo menos seis alunos terão a mesma nota. ◀

EXEMPLO 7

a) Quantas cartas devem ser escolhidas de um baralho de 52 cartas para garantir que pelo menos três cartas do mesmo naipe sejam selecionadas?

b) Quantas cartas devem ser escolhidas para garantir que pelo menos três copas sejam selecionadas?

Solução:
a) Suponha que haja quatro caixas, uma para cada naipe, e assim que as cartas forem sendo selecionadas, elas são colocadas na caixa reservada para cada naipe. Usando a generalização do princípio da casa dos pombos, vemos que se N cartas são selecionadas, há pelo menos uma caixa com no mínimo $\lceil N/4 \rceil$ cartas. Conseqüentemente, sabemos que pelo menos três cartas de um naipe são selecionadas se $\lceil N/4 \rceil \geq 3$. O menor número inteiro N, tal que $\lceil N/4 \rceil \geq 3$, é $N = 2 \cdot 4 + 1 = 9$, então nove cartas são suficientes. Note que se oito cartas forem escolhidas, é possível que haja duas cartas de cada naipe, então são necessárias mais que oito. Conseqüentemente, nove cartas devem ser selecionadas para garantir que pelo menos três cartas de um naipe sejam escolhidas. Uma boa forma de pensar nisso é notar que depois da oitava carta escolhida, não há como evitar ter uma terceira carta de algum naipe.
b) Não usaremos a generalização do princípio da casa dos pombos para responder a esta questão porque queremos ter certeza de que haja três copas, não apenas três cartas de um naipe. Note que, no pior dos casos, podemos selecionar todos os paus, ouros e espadas, 39 cartas no total, antes de escolher um único copas. As próximas três cartas serão todas copas, então precisaremos escolher 42 cartas para chegar nos três copas. ◀

EXEMPLO 8 Qual o menor número de códigos de área que são necessários para garantir que os 25 milhões de telefones em um estado possam ser determinados por números de telefones com 10 dígitos? (Considere que os números de telefone têm a forma *NXX-NXX-XXXX*, em que os três primeiros dígitos são o código de área, *N* representa um dígito entre 2 e 9 (inclusive) e *X* representa qualquer dígito.)

Solução: Há oito milhões de números de telefone diferentes na forma *NXX-XXXX* (como sabemos pelo Exemplo 8 da Seção 5.1). Portanto, pela generalização do princípio da casa dos pombos, entre 25 milhões de telefones, pelo menos $\lceil 25\,000\,000/8\,000\,000 \rceil$ deles devem ter números idênticos. Portanto, pelo menos quatro códigos de área são necessários para assegurar que todos os números de 10 dígitos sejam diferentes. ◀

O Exemplo 9, mesmo não sendo uma aplicação da generalização do princípio da casa dos pombos, usa princípios semelhantes.

EXEMPLO 9 Suponha que um laboratório de ciência da computação tenha 15 estações de trabalho e 10 servidores. Um cabo pode ser usado para conectar diretamente uma estação a um servidor. Para cada servidor, apenas uma conexão direta a este servidor pode ser ativada em qualquer hora. Queremos garantir que, a qualquer hora, qualquer conjunto de 10 ou menos estações de trabalho possa simultaneamente acessar servidores diferentes por meio de conexões diretas. Embora possamos fazer isto conectando cada estação diretamente a um servidor (usando 150 conexões), qual é o número mínimo de conexões diretas necessárias para alcançar este objetivo?

Solução: Suponha que rotulemos as estações de trabalho como $W_1, W_2, \ldots, W_{15}$ e os servidores como $S_1, S_2, \ldots, S_{10}$. Além disso, suponha que conectemos W_k a S_k para $k = 1, 2, \ldots, 10$ e cada um dos $W_{11}, W_{12}, W_{13}, W_{14}$ e W_{15} a todos os 10 servidores. Teremos um total de 60 conexões diretas. Claramente, qualquer conjunto de 10 ou menos estações de trabalho pode acessar simultaneamente servidores diferentes. Vemos isso a partir da observação de que se a estação de trabalho W_j estiver inclusa com $1 \leq j \leq 10$, ela poderá acessar o servidor S_j, e para cada estação W_k com $k \geq 11$ incluso, haverá uma estação correspondente W_j com $1 \leq j \leq 10$ não incluso, então W_k pode acessar o servidor S_j. (Isto é afirmado porque há pelo menos tantos servidores disponíveis S_j quanto estações W_j com $1 \leq j \leq 10$ não incluso.)

Agora suponha que há menos que 60 conexões diretas entre as estações e os servidores. Então, alguns servidores poderiam ser conectados a no máximo $\lfloor 59/10 \rfloor = 5$ estações. (Se todos os servidores forem conectados a pelo menos seis estações, então haveria no mínimo $6 \cdot 10 = 60$ conexões diretas.) Isto significa que os nove servidores restantes não são suficientes para permitir

que as outras 10 estações acessem simultaneamente servidores diferentes. Conseqüentemente, pelo menos 60 conexões diretas são necessárias. Segue-se que a resposta é 60. ◀

Algumas Aplicações Distintas do Princípio da Casa dos Pombos

Em muitas aplicações interessantes do princípio da casa dos pombos, os objetos que são colocados em caixas devem ser escolhidos de maneira engenhosa. Algumas dessas aplicações serão descritas aqui.

EXEMPLO 10 Durante um mês com 30 dias, um time de beisebol joga pelo menos uma partida por dia, mas não mais de 45 partidas. Mostre que deverá haver um período com alguns dias consecutivos em que o time joga exatamente 14 partidas.

Solução: Considere a_j como o número de partidas jogadas até o j-ésimo dia do mês. Então, $a_1, a_2, \ldots, a_{30}$ é uma seqüência crescente de números inteiros positivos distintos, com $1 \leq a_j \leq 45$. Além disso, $a_1 + 14, a_2 + 14, \ldots, a_{30} + 14$ é também uma seqüência crescente de números inteiros positivos distintos, com $15 \leq aj + 14 \leq 59$.

Os 60 números inteiros positivos $a_1, a_2, \ldots, a_{30}, a_1 + 14, a_2 + 14, \ldots, a_{30} + 14$ são todos menores ou iguais a 59. Assim, pelo princípio da casa dos pombos, dois desses números inteiros são iguais. Visto que os números inteiros $a_j, j = 1, 2, \ldots, 30$, são todos distintos e os números inteiros $a_j + 14, j = 1, 2, \ldots, 30$, também são todos distintos, deverá haver indícios de i e j com $a_i = a_j + 14$. Isto significa que exatamente 14 partidas foram jogadas do dia $j + 1$ ao dia i. ◀

EXEMPLO 11 Mostre que entre quaisquer $n + 1$ números inteiros positivos não excedentes a $2n$ deverá haver um número inteiro que divide um dos outros inteiros.

Solução: Escreva cada um dos $n + 1$ números inteiros $a_1, a_2, \ldots, a_{n+1}$ como uma potência de 2 vezes um inteiro ímpar. Em outras palavras, considere $a_j = 2^{k_j} q_j$ para $j = 1, 2, \ldots, n + 1$, em que k_j é um número inteiro não negativo e q_j é ímpar. Os inteiros $q_1, q_2, \ldots, q_{n+1}$ são todos números inteiros positivos ímpares menores que $2n$. Visto que há apenas n inteiros positivos ímpares menores que $2n$, podemos afirmar que, a partir do princípio da casa dos pombos, dois dos inteiros $q_1, q_2, \ldots, q_{n+1}$ deverão ser iguais. Então, há números inteiros i e j, tal que $q_i = q_j$. Considere q como o valor comum de q_i e q_j. Assim, $a_i = 2^{k_i} q$ e $a_j = 2^{k_j} q$. Concluímos que se $k_i < k_j$, então a_i divide a_j; enquanto se $k_i > k_j$, então a_j divide a_i. ◀

Uma aplicação engenhosa do princípio da casa dos pombos mostra a existência de uma subseqüência crescente ou decrescente de certa extensão em uma seqüência de números inteiros distintos. Algumas definições serão revisadas antes de esta aplicação ser apresentada. Suponha que $a_1, a_2, \ldots, a_N$ seja uma seqüência de números reais. Uma **subseqüência** desta seqüência é uma seqüência de forma $a_{i_1}, a_{i_2}, \cdots, a_{i_m}$, em que $1 \leq i_1 < i_2 < \cdots < i_m \leq N$. Assim, uma subseqüência é uma seqüência obtida a partir da seqüência original pela inclusão de alguns dos termos da seqüência original em sua ordem original. Uma seqüência é chamada de **estritamente crescente** se cada termo for maior que aquele que o precede, e é chamada de **estritamente decrescente** se cada termo for menor que aquele que o precede.

TEOREMA 3 Toda seqüência de $n^2 + 1$ números reais distintos contém uma subseqüência de extensão $n + 1$ que é estritamente crescente ou estritamente decrescente.

Daremos um exemplo antes de apresentar a demonstração do Teorema 3.

EXEMPLO 12 A seqüência 8, 11, 9, 1, 4, 6, 12, 10, 5, 7 contém 10 termos. Note que $10 = 3^2 + 1$. Há quatro subseqüências crescentes de extensão quatro, ou seja, 1, 4, 6, 12; 1, 4, 6, 7; 1, 4, 6, 10; e 1, 4, 5, 7. Há também uma subseqüência decrescente de extensão quatro, ou seja, 11, 9, 6, 5. ◀

A demonstração do teorema será agora apresentada.

Demonstração: Considere $a_1, a_2, \ldots, a_{n^2+1}$ como uma seqüência de $n^2 + 1$ números reais distintos. Associamos um par ordenado a cada termo da seqüência, ou seja, associamos (i_k, d_k) ao termo a_k, em que i_k é a extensão da maior subseqüência crescente, começando em a_k, e d_k é a extensão da maior subseqüência decrescente, começando em a_k.

Suponha que não haja subseqüências crescentes ou decrescentes de extensão $n + 1$. Então, i_k e d_k são números inteiros positivos menores que ou iguais a n, para $k = 1, 2, \ldots, n^2 + 1$. Assim, pela regra do produto, há n^2 pares ordenados possíveis para (i_k, d_k). Pelo princípio da casa dos pombos, dois desses $n^2 + 1$ pares ordenados são iguais. Em outras palavras, existem termos a_s e a_t, com $s < t$, tal que $i_s = i_t$ e $d_s = d_t$. Mostraremos que isto é impossível. Visto que os termos da seqüência são distintos, ou $a_s < a_t$ ou $a_s > a_t$. Se $a_s < a_t$, então, como $i_s = i_t$, pode ser construída uma subseqüência crescente de extensão $i_t + 1$ que começa em a_s, tomando a_s seguida por uma subseqüência crescente de extensão i_t, que começa em a_t. Isto é uma contradição. Analogamente, se $a_s > a_t$, pode ser mostrado que d_s deve ser maior que d_t, o que é uma contradição. ◁

Links O último exemplo mostra como a generalização do princípio da casa dos pombos pode ser aplicada a uma importante parte da análise combinatória chamada de **teoria de Ramsey**, criada pela matemático inglês F. P. Ramsey. Em geral, a teoria de Ramsey trabalha com a distribuição de subconjuntos de elementos de conjuntos.

EXEMPLO 13 Suponha que em um grupo de seis pessoas, cada par de indivíduos seja formado por dois amigos ou dois inimigos. Mostre que há três amigos mútuos ou três inimigos mútuos no grupo.

Solução: Considere A como uma das seis pessoas. Das cinco outras pessoas no grupo, há três ou mais que são amigos de A, ou três ou mais que são inimigos de A. Seguindo a generalização do princípio da casa dos pombos, quando cinco objetos são divididos em dois conjuntos, um dos conjuntos tem pelo menos $\lceil 5/2 \rceil = 3$ elementos. Neste caso, suponha que B, C e D são amigos de A. Se duas das três pessoas são amigas, então esses dois e A formam um grupo de três amigos mútuos. Por outro lado, B, C e D formam um conjunto de três inimigos mútuos. A demonstração neste segundo caso, quando há três ou mais inimigos de A, procede da mesma maneira. ◀

O **número de Ramsey** $R(m, n)$, em que m e n são números inteiros positivos maiores que ou iguais a 2, indica o número mínimo de pessoas em uma festa em que há m amigos mútuos ou n inimigos mútuos, supondo que todo par de pessoas na festa são amigos ou inimigos. O Exemplo 13 mostra que $R(3, 3) \leq 6$. Concluímos que $R(3, 3) = 6$ porque, em um grupo de cinco pessoas, em que cada duas pessoas são amigos ou inimigos, poderá não haver três amigos mútuos ou três inimigos mútuos (veja o Exercício 24).

Links

FRANK PLUMPTON RAMSEY (1903–1930) Frank Plumpton Ramsey, filho do presidente da Magdalene College, em Cambridge, foi educado em Winchester e Trinity Colleges. Depois de se formar em 1923, foi escolhido como membro da King's College, em Cambridge, onde passou o resto de sua vida. Ramsey fez importantes contribuições à lógica matemática. O que chamamos de teoria de Ramsey iniciou com seus argumentos combinatórios de grande habilidade, publicados no artigo "On a Problem of Formal Logic". Ramsey também fez contribuições à teoria matemática de economia. Ele era conhecido como um excelente professor em fundamentos da matemática. Sua morte, aos 26 anos de idade, privou a comunidade matemática e a Universidade de Cambridge de um jovem acadêmico brilhante.

É possível demonstrar algumas propriedades úteis sobre os números de Ramsey, mas, para a maior parte, é difícil encontrar seus valores exatos. Note que por simetria pode ser mostrado que $R(m, n) = R(n, m)$ (veja o Exercício 28). Temos também que $R(2, n) = n$ para todo número inteiro positivo $n \geq 2$ (veja o Exercício 27). São conhecidos apenas nove valores exatos dos números de Ramsey $R(m, n)$ com $3 \leq m \leq n$, incluindo $R(4, 4) = 18$. Apenas limites são conhecidos para muitos outros números de Ramsey, incluindo $R(5, 5)$, que é conhecido por satisfazer $43 \leq R(5, 5) \leq 49$. O leitor interessado em aprender mais sobre os números de Ramsey deverá consultar [MiRo91] ou [GrRoSp90].

Exercícios

1. Mostre que em qualquer conjunto de seis aulas, no qual cada encontro acontece regularmente uma vez na semana em um determinado dia da semana, deve haver dois desses encontros no mesmo dia, supondo que não haja aulas nos finais de semana.
2. Mostre que se há 30 estudantes em uma sala, então pelo menos dois têm sobrenomes que começam com a mesma letra.
3. Uma gaveta contém doze meias marrons e doze meias pretas, todas únicas. Um homem pega as meias aleatoriamente no escuro.
   a) Quantas meias deverão ser pegas para se ter certeza de que pelo menos duas são da mesma cor?
   b) Quantas meias deverão ser pegas para se ter certeza de que pelo menos duas são pretas?
4. Um boliche contém 10 bolas vermelhas e 10 bolas azuis. Uma mulher escolhe as bolas aleatoriamente sem olhar para elas.
   a) Quantas bolas deverão ser pegas para se ter certeza de que pelo menos três bolas são da mesma cor?
   b) Quantas bolas deverão ser pegas para se ter certeza de que pelo menos três são azuis?
5. Mostre que entre um grupo de cinco números inteiros (não necessariamente consecutivos) há dois com o mesmo resto quando divididos por 4.
6. Considere d como um número inteiro positivo. Mostre que entre um grupo de $d + 1$ números inteiros (não necessariamente consecutivos) há dois com exatamente o mesmo resto quando divididos por d.
7. Considere n como um número inteiro positivo. Mostre que em um conjunto de n inteiros consecutivos há exatamente um número divisível por n.
8. Mostre que se f for uma função de S para T, em que S e T são conjuntos finitos com $|S| > |T|$, então há elementos s_1 e s_2 em S, tal que $f(s_1) = f(s_2)$, ou, em outras palavras, f não é injetora.
9. Qual é o número mínimo de estudantes, vindos dos 50 estados americanos, que devem ser inscritos em uma universidade para garantir que pelo menos 100 venham do mesmo estado?
10. Considere (x_i, y_i), $i = 1, 2, 3, 4, 5$, como um conjunto de cinco pontos distintos com coordenadas inteiras no plano xy. Mostre que o ponto médio da linha que junta pelo menos um par desses pontos tem coordenadas inteiras.
11. Considere (x_i, y_i, z_i), $i = 1, 2, 3, 4, 5, 6, 7, 8, 9$, um conjunto de nove pontos distintos com coordenadas inteiras no espaço xyz. Mostre que o ponto médio de pelo menos um par desses pontos tem coordenadas inteiras.
12. Quantos pares ordenados de números inteiros (a, b) são necessários para garantir que haja dois pares ordenados (a_1, b_1) e (a_2, b_2), tal que $a_1 \bmod 5 = a_2 \bmod 5$ e $b_1 \bmod 5 = b_2 \bmod 5$?
13. a) Mostre que se cinco números inteiros são selecionados a partir dos oito primeiros números inteiros positivos, deverá haver um par desses inteiros cuja soma seja igual a 9.
    b) A conclusão do item (a) será verdadeira se quatro inteiros, em vez de cinco, forem selecionados?
14. a) Mostre que se sete números inteiros são selecionados a partir dos 10 primeiros números inteiros positivos, deverá haver pelo menos dois pares desses inteiros cuja soma seja igual a 11.
    b) A conclusão do item (a) será verdadeira se seis inteiros, em vez de sete, forem selecionados?
15. Quantos números devem ser selecionados a partir do conjunto $\{1, 2, 3, 4, 5, 6\}$ para garantir que pelo menos um par desses números some 7?
16. Quantos números devem ser selecionados a partir do conjunto $\{1, 3, 5, 7, 9, 11, 13, 15\}$ para garantir que pelo menos um par desses números some 16?
17. Uma companhia armazena os produtos em um depósito. O armazenamento de caixas nesse depósito é especificado por seus corredores, localização nos corredores e prateleiras. Há 50 corredores, 85 localizações horizontais em cada corredor e 5 prateleiras no depósito. Qual é o menor número de produtos que a companhia pode ter para que pelo menos dois produtos sejam estocados na mesma caixa?
18. Suponha que haja nove estudantes em uma classe de matemática discreta em uma pequena faculdade.
    a) Mostre que a classe deve ter pelo menos cinco homens ou pelo menos cinco mulheres.
    b) Mostre que a classe deve ter pelo menos três homens ou pelo menos sete mulheres.
19. Suponha que todo estudante em uma classe de matemática discreta de 25 alunos seja um calouro, um veterano ou um aluno júnior.
    a) Mostre que há pelo menos nove calouros, nove veteranos ou nove juniores na sala.

b) Mostre que há pelo menos três calouros, 19 veteranos ou 5 juniores na sala.

20. Encontre uma subseqüência crescente de extensão máxima e uma subseqüência decrescente de extensão máxima a partir da seqüência 22, 5, 7, 2, 23, 10, 15, 21, 3, 17.

21. Construa uma seqüência de 16 números inteiros positivos que não tenha subseqüência crescente ou decrescente de cinco termos.

22. Mostre que se houver 101 pessoas de alturas diferentes paradas em fila, é possível encontrar 11 pessoas na ordem que estão paradas em uma seqüência de alturas crescente ou decrescente.

***23.** Descreva um algoritmo em pseudocódigo para produzir a maior subseqüência crescente ou decrescente de uma seqüência de números inteiros distintos.

24. Mostre que em um grupo de cinco pessoas (em que duas pessoas são amigas ou inimigas) não há necessariamente três amigos mútuos ou três inimigos mútuos.

25. Mostre que em um grupo de 10 pessoas (em que duas pessoas são amigas ou inimigas) há três amigos mútuos ou quatro inimigos mútuos, e que há três inimigos mútuos ou quatro amigos mútuos.

26. Use o Exercício 25 para mostrar que entre um grupo de 20 pessoas (em que duas pessoas são amigas ou inimigas) há quatro amigos mútuos ou quatro inimigos mútuos.

27. Mostre que se n for um número inteiro positivo com $n \geq 2$, então o número de Ramsey $R(2, n)$ é igual a n.

28. Mostre que se m e n forem números inteiros positivos com $m \geq 2$ e $n \geq 2$, então os números de Ramsey $R(m, n)$ e $R(n, m)$ são iguais.

29. Mostre que há pelo menos seis pessoas na Califórnia (população: 36 milhões) com as mesmas três iniciais que nasceram no mesmo dia do ano (mas não necessariamente no mesmo ano). Suponha que todos tenham três iniciais.

30. Mostre que se houver 100 000 000 pessoas no Brasil que ganham salário menores que 1 000 000 reais, então há duas pessoas que ganharam exatamente a mesma quantia, incluindo os centavos, no ano passado.

31. Há 38 horários diferentes de aula em uma universidade. Se há 677 classes diferentes, quantas salas diferentes serão necessárias?

32. Uma rede de computadores é formada por seis computadores. Cada computador é conectado diretamente a, pelo menos, um dos outros computadores. Mostre que há pelo menos dois computadores na rede que estão diretamente conectados ao mesmo número de outros computadores.

33. Uma rede de computadores é formada por seis computadores. Cada computador é conectado diretamente a zero ou mais outros computadores. Mostre que há pelo menos dois computadores na rede que estão diretamente conectados ao mesmo número de outros computadores. [*Dica:* É impossível ter um computador não ligado a nenhum dos outros e um computador ligado a todos os outros.]

34. Encontre o menor número de cabos que são necessários para conectar oito computadores a quatro impressoras a fim de garantir que quatro computadores estejam ligados diretamente a quatro impressoras diferentes. Justifique sua resposta.

35. Encontre o menor número de cabos que são necessários para conectar 100 computadores a 20 impressoras para garantir que 20 computadores possam acessar diretamente 20 impressoras diferentes. (Aqui, as hipóteses sobre os cabos e computadores são as mesmas que as do Exemplo 9.) Justifique sua resposta.

***36.** Demonstre que em uma festa em que há pelo menos duas pessoas, há duas pessoas que conhecem o mesmo número de pessoas na festa.

37. Um campeonato de luta dura 75 horas. (Aqui, por uma hora entendemos um período que começa em uma hora exata, como 13h, e vai até a próxima hora.) O lutador teve pelo menos uma competição por hora, mas não mais que um total de 125 competições. Mostre que há um período de horas consecutivas em que o lutador teve exatamente 24 competições.

***38.** A proposição do Exercício 37 será verdadeira se 24 for substituído por

a) 2?

b) 23?

c) 25?

d) 30?

39. Mostre que se f for uma função de S para T, em que S e T são conjuntos finitos e $m = \lceil |S|/|T| \rceil$, então pelo menos m elementos de S têm a mesma imagem em T. Ou seja, mostre que há elementos distintos $s_1, s_2, \ldots, s_m$ de S tal que $f(s_1) = f(s_2) = \cdots = f(s_m)$.

40. Há 51 casas em uma rua. Cada casa tem um número entre 1000 e 1099, inclusive. Mostre que pelo menos duas casas têm números que são inteiros consecutivos.

***41.** Considere x como um número irracional. Mostre que para um número inteiro positivo j não excedente a n, o valor absoluto da diferença entre $j\,x$ e o inteiro mais próximo a $j\,x$ é menor que $1/n$.

42. Considere $n_1, n_2, \ldots, n_t$ como números inteiros positivos. Mostre que se $n_1 + n_2 + \cdots + n_t - t + 1$ objetos forem colocados em t caixas, então para i, $i = 1, 2, \ldots, t$, a i-ésima caixa terá pelo menos n_i objetos.

***43.** Uma demonstração do Teorema 3 baseada na generalização do princípio da casa dos pombos é apresentada neste exercício. A notação usada é a mesma da demonstração no texto.

a) Suponha que $i_k \leq n$ para $k = 1, 2, \ldots, n^2 + 1$. Use a generalização do princípio da casa dos pombos para mostrar que há $n + 1$ termos $a_{k_1}, a_{k_2}, \cdots, a_{k_{n+1}}$ com $i_{k_1} = i_{k_2} = \cdots = i_{k_{n+1}}$, em que $1 \leq k_1 < k_2 < \cdots < k_{n+1}$.

b) Mostre que $a_{k_j} > a_{k_{j+1}}$ para $j = 1, 2, \ldots, n$. [*Dica:* Suponha que $a_{k_j} < a_{k_{j+1}}$, e mostre que isto implica que $i_{k_j} > i_{k_{j+1}}$, o que é uma contradição.]

c) Use os itens (a) e (b) para mostrar que se não houver subseqüência crescente de extensão $n + 1$, então deverá haver uma subseqüência decrescente com a mesma extensão.

5.3 Permutações e Combinações

Introdução

Muitos problemas de contagem podem ser resolvidos encontrando-se o número de maneiras de organizar um número específico de elementos distintos de um conjunto de determinado tamanho, em que a ordem desses elementos é relevante. Muitos outros problemas de contagem podem ser resolvidos encontrando-se o número de maneiras para escolher um número determinado de elementos a partir de um conjunto de determinado tamanho, em que a ordem dos elementos escolhidos não é relevante. Por exemplo, de quantas maneiras podemos escolher três alunos, de um grupo de cinco estudantes, para ficarem em fila para uma foto? Quantos comitês diferentes de três estudantes podem ser formados a partir de um grupo com quatro estudantes? Nesta seção desenvolveremos métodos para responder a tais questões.

Permutações

Começaremos pela resolução da primeira questão proposta na introdução desta seção, assim como das questões relacionadas a ela.

EXEMPLO 1 De quantas maneiras podemos escolher três alunos, de um grupo de cinco estudantes, para ficarem em fila para uma foto? De quantas maneiras podemos organizar todos os cinco estudantes em fila para uma foto?

Exemplos Extras

Solução: Primeiro, note que a ordem na qual selecionamos os estudantes é relevante. Há cinco maneiras de escolher o primeiro estudante para ficar no começo da fila. Uma vez escolhido este estudante, há quatro maneiras de escolher o segundo da fila. Depois do primeiro e do segundo estudante escolhidos, há três maneiras de escolher o terceiro estudante na fila. Pela regra do produto, há $5 \cdot 4 \cdot 3 = 60$ maneiras de escolher três estudantes de um grupo de cinco para ficarem em fila para uma foto.

Para organizar os cinco estudantes em uma fila para uma foto, temos cinco formas para selecionar o primeiro, quatro para o segundo, três para o terceiro, duas para o quarto e uma para o quinto. Conseqüentemente, temos $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$ maneiras de organizar os cinco estudantes em uma fila para uma foto. ◀

O Exemplo 1 mostra como arranjos ordenados de objetos distintos podem ser contados. Isto nos conduz a algumas terminologias.

Uma **permutação** de um conjunto de objetos distintos é um arranjo ordenado desses objetos. Interessamo-nos também por arranjos ordenados de alguns dos elementos de um conjunto. Um arranjo ordenado de r elementos de um conjunto é chamado de ***r*-permutação.**

Links

EXEMPLO 2 Considere $S = \{1, 2, 3\}$. O arranjo ordenado 3, 1, 2 é uma permutação de S. O arranjo ordenado 3, 2 é uma 2-permutação de S. ◀

O número de r-permutações de um conjunto com n elementos é indicado por $P(n, r)$. Podemos encontrar $P(n, r)$ usando a regra do produto.

EXEMPLO 3 Considere $S = \{a, b, c\}$. A 2-permutação de S são arranjos ordenados a, b; a, c; b, a; b, c; c, a e c, b. Conseqüentemente, há seis 2-permutações deste conjunto com três elementos. Para ver que há sempre seis 2-permutações de um conjunto com três elementos, note que há três maneiras de escolher o primeiro elemento de um arranjo e duas maneiras de escolher o segundo elemento do arranjo, porque este deve ser diferente do primeiro. Pela regra do produto, temos que $P(3, 2) = 3 \cdot 2 = 6$. ◀

Usaremos agora a regra do produto para encontrar a fórmula para $P(n, r)$, sempre que n e r forem números inteiros positivos com $1 \leq r \leq n$.

TEOREMA 1 Se n for um número inteiro positivo e r um número inteiro com $1 \leq r \leq n$, então há

$$P(n,r) = n(n-1)(n-2)\cdots(n-r+1)$$

r-permutações de um conjunto com n elementos distintos.

Demonstração: Usaremos a regra do produto para demonstrar que esta fórmula está correta. O primeiro elemento da permutação pode ser escolhido de n maneiras, porque há n elementos no conjunto. Há $n-1$ maneiras de escolher o segundo elemento da permutação, porque há $n-1$ elementos que sobraram no conjunto depois de utilizar o elemento da primeira posição. Do mesmo modo, há $n-2$ maneiras de escolher o terceiro elemento, e assim por diante, até que tenha exatamente $n-(r-1) = n-r+1$ maneiras de escolher o r-ésimo elemento. Conseqüentemente, pela regra do produto, há

$$n(n-1)(n-2)\cdots(n-r+1)$$

r –permutações do conjunto. ◁

Note que $P(n, 0) = 1$ sempre que n for um número inteiro não negativo, porque há exatamente uma maneira de ordenar zero elementos. Isto é, há exatamente uma lista sem elementos, ou seja, uma lista vazia.

Apresentaremos agora um corolário muito útil do Teorema 1.

COROLÁRIO 1 Se n e r são números inteiros com $0 \leq r \leq n$, então $P(n,r) = \dfrac{n!}{(n-r)!}$

Demonstração: Quando n e r são números inteiros com $1 \leq r \leq n$, pelo Teorema 1 temos

$$P(n,r) = n(n-1)(n-2)\cdots(n-r+1) = \frac{n!}{(n-r)!}$$

Como $\dfrac{n!}{(n-0)!} = \dfrac{n!}{n!} = 1$, sempre que n for um número inteiro não negativo, vemos que aquela fórmula $P(n,r) = \dfrac{n!}{(n-r)!}$ também é válida para quando $r = 0$. ◁

Pelo Teorema 1, sabemos que se n for um número inteiro positivo, então $P(n, n) = n!$. Mostraremos este resultado com alguns exemplos.

EXEMPLO 4 Quantas maneiras há para selecionar o vencedor do primeiro, do segundo e do terceiro lugar a partir de 100 pessoas diferentes que participaram de um concurso?

Solução: Como a ordem é relevante, já que cada pessoa ganha um prêmio, o número de maneiras de selecionar os três vencedores é o número de escolhas ordenadas de três elementos de um grupo de 100 elementos, isto é, o número de 3-permutações de um conjunto de 100 elementos. Conseqüentemente, a resposta será

$$P(100, 3) = 100 \cdot 99 \cdot 98 = 970\,200.$$ ◀

EXEMPLO 5 Suponha que haja oito atletas em uma corrida. O vencedor recebe medalha de ouro, o segundo lugar, medalha de prata e o terceiro, medalha de bronze. Quantas maneiras diferentes há para ganhar essas medalhas, se todos os resultados possíveis da corrida podem ocorrer e não há nenhum empate?

Solução: O número de maneiras diferentes de distribuir as medalhas é o número de 3-permutações de um conjunto com oito elementos. Assim, há $P(8, 3) = 8 \cdot 7 \cdot 6 = 336$ maneiras possíveis de ganhar as medalhas. ◀

EXEMPLO 6 Suponha que uma vendedora tenha de visitar oito cidades diferentes. Ela deve começar sua viagem por uma determinada cidade, mas depois pode visitar as outras sete na ordem em que ela quiser. Quantas ordens são possíveis para a vendedora visitar as cidades?

Solução: O número de caminhos possíveis entre as cidades é o número de permutações de sete elementos, porque a primeira cidade é determinada, mas as sete restantes podem ser ordenadas arbitrariamente. Conseqüentemente, há $7! = 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 5040$ maneiras para a vendedora escolher seu roteiro. Se, por exemplo, ela desejar encontrar um caminho entre as cidades que tenha a menor distância, deve tomar a distância total para cada caminho possível e deve considerar um total de 5040 caminhos! ◀

EXEMPLO 7 Quantas permutações das letras *ABCDEFGH* contêm a seqüência *ABC*?

Solução: Visto que as letras *ABC* devem ser consideradas como um bloco, podemos resolver esta questão encontrando o número de permutações de seis objetos, ou seja, o bloco *ABC* e as letras individuais *D*, *E*, *F*, *G* e *H*. Como os seis objetos podem aparecer em qualquer ordem, há $6! = 720$ permutações das letras *ABCDEFGH* nas quais *ABC* aparecem como um bloco. ◀

Combinações

Agora estudaremos a seleção não ordenada de objetos. Comecemos pela resolução de uma questão proposta na introdução desta seção.

EXEMPLO 8 Quantos comitês diferentes de três estudantes podem ser formados a partir de um grupo de quatro estudantes?

Solução: Para responder a esta questão, precisamos apenas encontrar o número de subconjuntos com três elementos a partir de um conjunto com os quatro estudantes. Vemos que há quatro subconjuntos, um para cada um dos quatro estudantes, já que escolher quatro deles é o mesmo que escolher um para ser omitido do grupo. Isto significa que há quatro maneiras de escolher três alunos para o comitê, em que a ordem de escolha desses estudantes não é relevante. ◀

O Exemplo 8 mostra que muitos problemas de contagem podem ser resolvidos encontrando-se o número de subconjuntos de um determinado tamanho a partir de um conjunto com n elementos, em que n é um número inteiro positivo.

Uma **r-combinação** de elementos de um conjunto é uma seleção não ordenada de r elementos a partir do conjunto. Então, uma r-combinação é simplesmente um subconjunto do conjunto com r elementos.

EXEMPLO 9 Considere S como o conjunto $\{1, 2, 3, 4\}$. Então, $\{1, 3, 4\}$ é uma 3-combinação de S. ◀

O número de r-combinações de um conjunto com n elementos distintos é representado por $C(n, r)$. Note que $C(n, r)$ é também representado por $\binom{n}{r}$ e é chamado de **coeficiente binomial.** Aprenderemos a origem dessa terminologia na Seção 5.4.

EXEMPLO 10 Vemos que $C(4, 2) = 6$, porque as 2-combinações de $\{a, b, c, d\}$ são os seis subconjuntos $\{a,b\}$, $\{a, c\}$, $\{a, d\}$, $\{b, c\}$, $\{b, d\}$ e $\{c, d\}$. ◀

Podemos determinar o número de r-combinações de um conjunto com n elementos usando a fórmula para o número de r-permutações de um conjunto. Para fazer isso, note que as r-permutações de um conjunto podem ser obtidas a partir, primeiramente, da formação de r-combinações e, depois, pelo ordenamento dos elementos nessas combinações. A demonstração do Teorema 2, que apresenta o valor de $C(n, r)$, é baseada nesta observação.

TEOREMA 2 O número de r-combinações de um conjunto com n elementos, em que n é um número inteiro não negativo e r é um inteiro com $0 \leq r \leq n$, é igual a

$$C(n,r) = \frac{n!}{r!(n-r)!}.$$

Demonstração: As r-permutações de um conjunto podem ser obtidas a partir da formação de r-combinações $C(n, r)$ do conjunto e, então, pelo ordenamento desses elementos em cada r-combinação, o que pode ser feito de $P(r, r)$ maneiras. Conseqüentemente,

$$P(n, r) = C(n, r) \cdot P(r, r).$$

Isto implica que

$$C(n,r) = \frac{P(n,r)}{P(r,r)} = \frac{n!/(n-r)!}{r!/(r-r)!} = \frac{n!}{r!(n-r)!}.$$

◁

A fórmula no Teorema 2, embora explícita, não é útil quando $C(n, r)$ é computado com valores altos para n e r. A razão para tal afirmação é que computar valores exatos de fatoriais é interessante apenas em números inteiros pequenos, e quando a aritmética de ponto flutuante é usada, a fórmula do Teorema 2 pode produzir um valor que não é inteiro. Ao computar $C(n, r)$, primeiro note que quando eliminamos $(n - r)!$ do numerador e do denominador da expressão para $C(n, r)$ no Teorema 2, obtemos

$$C(n, r) = \frac{n!}{r!(n-r)!} = \frac{n(n-1)\cdots(n-r+1)}{r!}.$$

Conseqüentemente, para computar $C(n, r)$ você pode eliminar todos os termos no fatorial maior no denominador a partir do numerador e do denominador, então multiplicar todos os termos que não forem cancelados no numerador e, por fim, dividir pelo menor fatorial no denominador. [Quando fazemos este cálculo manualmente, em vez de utilizar a calculadora, vale a pena trabalhar com fatores comuns no numerador $n(n-1)\cdots(n-r+1)$ e no denominador r!.] Note que muitas calculadoras têm uma função para $C(n, r)$ que pode ser usada para valores relativamente pequenos de n e r e muitos programas computacionais podem ser usados para encontrar $C(n, r)$. [Tais funções podem ser chamadas de *chose*(n, k) ou *binom*(n, k)].

O Exemplo 11 mostra como $C(n, k)$ é computado quando k é relativamente pequeno, comparado a n, e quando k é próximo de n. Mostra também uma identidade-chave dos números $C(n, k)$.

EXEMPLO 11 Quantas mãos de pôquer de cinco cartas podem ser retiradas a partir de um baralho de 52 cartas? Quantas formas de selecionar 47 cartas a partir de um baralho de 52 cartas são possíveis?

Solução: Visto que a ordem com que as cinco cartas são retiradas de um baralho de 52 cartas não é relevante, há

$$C(52, 5) = \frac{52!}{5!47!}$$

mãos diferentes de cinco cartas que podem ser tiradas. Para computar o valor de $C(52, 5)$, primeiro divide-se o numerador e o denominador por 47! para obter

$$C(52, 5) = \frac{52 \cdot 51 \cdot 50 \cdot 49 \cdot 48}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}.$$

Esta expressão pode ser simplificada dividindo-se 50 por 5 do denominador para obter um fator 10 no numerador, então dividir 48 por 4 para obter um fator 12 no numerador, depois dividir 51 por 3 e obter um fator 17 no numerador e, finalmente, dividir 52 por 2 para obter um fator 26 no numerador. Encontramos

$$C(52, 5) = 26 \cdot 17 \cdot 10 \cdot 49 \cdot 12 = 2\,598\,960.$$

Conseqüentemente, há 2 598 960 mãos de pôquer diferentes de cinco cartas que podem ser tiradas a partir de um baralho com 52 cartas.

Note que há

$$C(52, 47) = \frac{52!}{47!5!}$$

maneiras diferentes de selecionar 47 cartas de um baralho com 52. Mas não precisamos computar esse valor, porque $C(52, 47) = C(52, 5)$. (Apenas a ordem dos fatores 5! e 47! é diferente nos denominadores nas fórmulas para essas quantias.) Vemos que há também 2 598 960 maneiras diferentes de selecionar 47 cartas a partir de um baralho com 52 cartas. ◀

No Exemplo 11, observamos que $C(52, 5) = C(52, 47)$. Este caso especial de identidade útil para os números de r-combinações de um conjunto é dado no Corolário 2.

COROLÁRIO 2 Considere n e r como números inteiros não negativos com $r \leq n$. Então $C(n, r) = C(n, n - r)$.

Demonstração: A partir do Teorema 2, temos que

$$C(n, r) = \frac{n!}{r!(n - r)!}$$

e

$$C(n, n - r) = \frac{n!}{(n - r)![n - (n - r)]!} = \frac{n!}{(n - r)!r!}.$$

Assim, $C(n, r) = C(n, n - r)$. ◁

Podemos também demonstrar o Corolário 2 usando uma demonstração que mostra que os dois lados da equação no Corolário 2 fazem a contagem dos mesmos objetos usando razões diferentes. Descrevemos este importante tipo de demonstração na Definição 1.

DEFINIÇÃO 1 Uma *demonstração combinatória* de uma identidade é aquela que usa argumentos de contagem para demonstrar que os dois lados de uma identidade fazem a contagem dos mesmos objetos, mas de maneiras diferentes.

Muitas identidades que envolvem coeficientes binomiais podem ser demonstradas por meio das demonstrações combinatórias ou combinatoriais. Fornecemos agora uma demonstração combinatória do Corolário 2.

Demonstração: Suponha que S seja um conjunto com n elementos. Cada subconjunto A de S com r elementos corresponde a um subconjunto de S com $n - r$ elementos, ou seja, $\overline{A}$. Conseqüentemente, $C(n, r) = C(n, n - r)$. ◁

EXEMPLO 12 Quantas maneiras há para selecionar cinco jogadores de um time de basquete com 10 membros para disputar uma partida em outra escola?

Exemplos Extras

Solução: A resposta é dada pelo número de 5-combinações de um conjunto com 10 elementos. Pelo Teorema 2, o número de tais combinações é

$$C(10, 5) = \frac{10!}{5!5!} = 252.$$ ◀

EXEMPLO 13 Um grupo de 30 pessoas foi treinado a fim de serem formados astronautas para ir a uma primeira missão a Marte. Quantas maneiras de selecionar uma tripulação de seis pessoas para embarcarem nessa missão são possíveis? (Suponha que todos os membros da tripulação têm a mesma função.)

Solução: O número de maneiras de selecionar uma tripulação de seis pessoas a partir de um grupo de 30 é o número de 6-combinações de um conjunto de 30 elementos, porque a ordem dessas pessoas escolhidas não é relevante. Pelo Teorema 2, o número de tais combinações é

$$C(30, 6) = \frac{30!}{6!24!} = \frac{30 \cdot 29 \cdot 28 \cdot 27 \cdot 26 \cdot 25}{6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} = 593\,775.$$ ◀

EXEMPLO 14 Quantas cadeias de bits de extensão n contêm exatamente r 1s?

Solução: As posições de r 1s em uma cadeia de bits de extensão n formam uma r-combinação do conjunto com $\{1, 2, 3, \ldots, n\}$. Assim, há $C(n, r)$ cadeias de bits de extensão n que contêm exatamente r 1s. ◀

EXEMPLO 15 Suponha que haja 9 membros da faculdade no departamento de matemática e 11 no departamento de ciência da computação. Quantas maneiras de selecionar um comitê para desenvolver um curso de matemática discreta na escola são possíveis, se o comitê é formado por três membros do departamento de matemática e quatro do departamento de ciência da computação?

Solução: Pela regra do produto, a resposta é o produto dos números de 3-combinações de um conjunto com nove elementos e o número de 4-combinações de um conjunto com 11 elementos. Pelo Teorema 2, o número de maneiras de selecionar o comitê é

$$C(9, 3) \cdot C(11, 4) = \frac{9!}{3!6!} \cdot \frac{11!}{4!7!} = 84 \cdot 330 = 27\,720.$$ ◀

Exercícios

1. Liste todas as permutações de $\{a, b, c\}$.
2. Quantas permutações diferentes são possíveis a partir do conjunto $\{a, b, c, d, e, f, g\}$?
3. Quantas permutações de $\{a, b, c, d, e, f, g\}$ terminam com a?
4. Considere $S = \{1, 2, 3, 4, 5\}$.
   a) Liste todas as 3-permutações de S.
   b) Liste todas as 3-combinações de S.
5. Encontre o valor para cada uma das quantidades abaixo.
   a) $P(6, 3)$ **b)** $P(6, 5)$

c) $P(8, 1)$ **d)** $P(8, 5)$
e) $P(8, 8)$ **f)** $P(10, 9)$

6. Encontre o valor para cada uma das quantidades abaixo.
a) $C(5, 1)$ **b)** $C(5, 3)$
c) $C(8, 4)$ **d)** $C(8, 8)$
e) $C(8, 0)$ **f)** $C(12, 6)$

7. Encontre o número de 5-permutações de um conjunto com nove elementos.

8. Em quantas ordens diferentes cinco atletas podem terminar uma corrida sem empates?

9. Quantas possibilidades há para as posições de primeiro, segundo e terceiro lugares em uma corrida de cavalos com 12 cavalos, se todas as ordens de término são possíveis?

10. Há seis candidatos diferentes para governador de um estado. De quantas maneiras diferentes os nomes dos candidatos podem ser impressos na cédula de voto?

11. Quantas cadeias de bits de extensão 10 contêm
a) exatamente quatro 1s?
b) no máximo quatro 1s?
c) pelo menos quatro 1s?
d) um número igual de 0s e 1s?

12. Quantas cadeias de bits de extensão 12 contêm
a) exatamente três 1s?
b) no máximo três 1s?
c) pelo menos três 1s?
d) um número igual de 0s e 1s?

13. Um grupo contém n homens e n mulheres. Há quantas maneiras possíveis de organizar essas pessoas em uma fila se os homens e as mulheres devem ficar alternados?

14. De quantas maneiras um conjunto de dois números inteiros positivos, menores que 100, podem ser escolhidos?

15. De quantas maneiras um conjunto com cinco letras pode ser selecionado a partir do alfabeto da língua inglesa?

16. Um conjunto de 10 elementos pode ter quantos subconjuntos com um número ímpar de elementos?

17. Um conjunto com 100 elementos pode ter quantos subconjuntos com mais de dois elementos?

18. Uma moeda é jogada oito vezes, e em cada lançamento tem-se cara ou coroa. Quantos resultados são possíveis
a) no total?
b) com exatamente três caras?
c) com pelo menos três caras?
d) com o mesmo número de caras e coroas?

19. Uma moeda é jogada 10 vezes, e em cada lançamento tem-se cara ou coroa. Quantos resultados são possíveis
a) no total?
b) com exatamente duas caras?
c) com no máximo três coroas?
d) com o mesmo número de caras e coroas?

20. Quantas cadeias de bits de extensão 10 têm
a) exatamente três 0s?
b) mais 0s que 1s?
c) pelo menos sete 1s?
d) pelo menos três 1s?

21. Quantas permutações das letras *ABCDEFG* contêm
a) a seqüência *BCD*?
b) a seqüência *CFGA*?
c) as seqüências *BA* e *GF*?
d) as seqüências *ABC* e *DE*?
e) as seqüências *ABC* e *CDE*?
f) as seqüências *CBA* e *BED*?

22. Quantas permutações das letras *ABCDEFGH* contêm
a) a seqüência *ED*?
b) a seqüência *CDE*?
c) as seqüências *BA* e *FGH*?
d) as seqüências *AB*, *DE* e *GH*?
e) as seqüências *CAB* e *BED*?
f) as seqüências *BCA* e *ABF*?

23. Há quantas maneiras possíveis para que oito homens e cinco mulheres fiquem em uma fila, de modo que haja duas mulheres uma ao lado da outra? [*Dica:* Primeiro posicione os homens e depois considere as posições possíveis para as mulheres.]

24. Há quantas maneiras possíveis para que 10 mulheres e seis homens fiquem em uma fila, de forma que não haja dois homens um ao lado do outro? [*Dica:* Primeiro posicione as mulheres e depois considere as posições possíveis para os homens.]

25. Cem bilhetes, numerados de 1, 2, 3, . . . , 100, são vendidos a 100 pessoas diferentes para uma atração. Quatro prêmios diferentes são disputados, inclusive o grande prêmio (uma viagem para o Taiti). Há quantas maneiras possíveis de ganhar os prêmios, se
a) não há restrições?
b) a pessoa com o bilhete 47 ganhar o grande prêmio?
c) a pessoa com o bilhete 47 ganhar um dos prêmios?
d) a pessoa com o bilhete 47 não ganhar um prêmio?
e) as pessoas com os bilhetes 19 e 47 ganharem prêmios?
f) as pessoas com os bilhetes 19, 47 e 73 ganharem prêmios?
g) as pessoas com os bilhetes 19, 47, 73 e 97 ganharem prêmios?
h) nenhuma das pessoas com os bilhetes 19, 47, 73 e 97 ganharem prêmios?
i) o vencedor do grande prêmio for uma pessoa com o bilhete 19, 47, 73 ou 97?
j) as pessoas com os bilhetes 19 e 47 ganharem prêmios, mas as pessoas com os bilhetes 73 e 97 não ganharem?

26. Treze pessoas de um time de softball aparecem para um jogo.
a) Quantas maneiras de escolher 10 jogadores para entrar em campo são possíveis?
b) Quantas maneiras de designar as 10 posições são possíveis, selecionando-se os jogadores a partir das 13 pessoas que apareceram?

c) Das 13 pessoas que apareceram, três são mulheres. Há quantas maneiras possíveis de escolher os 10 jogadores se pelo menos um desses jogadores deve ser uma mulher?

27. Um clube tem 25 membros.

a) Quantas maneiras de escolher quatro membros do clube para servir no comitê executivo são possíveis?

b) Quantas maneiras de escolher um presidente, vice-presidente, secretário e tesoureiro do clube, em que nenhuma pessoa pode ter mais que uma função, são possíveis?

28. Um professor escreve 40 questões do tipo falso/verdadeiro de matemática discreta. Das proposições dessas questões, 17 são verdadeiras. Se as questões podem ser organizadas em qualquer ordem, quantos cadernos de respostas diferentes são possíveis?

***29.** Quantas 4-permutações de números inteiros positivos, não excedentes a 100, contêm três números inteiros consecutivos $k, k+1, k+2$, na ordem correta

a) em que esses números inteiros consecutivos podem talvez ser separados por outros inteiros na permutação?

b) em que eles estão em posições consecutivas na permutação?

30. Há sete mulheres e nove homens em uma faculdade no departamento de matemática.

a) Quantas maneiras de selecionar um comitê de cinco membros do departamento são possíveis, se pelo menos uma mulher deve estar no comitê?

b) Quantas maneiras de selecionar um comitê de cinco membros do departamento são possíveis, se pelo menos uma mulher e um homem devem estar no comitê?

31. O alfabeto da língua inglesa contém 21 consoantes e cinco vogais. Quantas seqüências de seis letras minúsculas do alfabeto contêm

a) exatamente uma vogal?

b) exatamente duas vogais?

c) pelo menos uma vogal?

d) pelo menos duas vogais?

32. Quantas seqüências de seis letras minúsculas do alfabeto da língua inglesa contêm

a) a letra a?

b) as letras a e b?

c) as letras a e b em posições consecutivas, com a precedendo b, com todas as letras distintas?

d) as letras a e b, em que a está à esquerda de b em uma seqüência, com todas as letras distintas?

33. Suponha que um departamento tenha 10 homens e 15 mulheres. Quantas maneiras de formar um comitê com seis membros são possíveis, se ele deve ter o mesmo número de homens e mulheres?

34. Suponha que um departamento tenha 10 homens e 15 mulheres. Quantas maneiras de formar um comitê com seis membros são possíveis, se ele deve ter mais mulheres que homens?

35. Quantas cadeias de bits contêm exatamente oito 0s e dez 1s, se todo 0 deve ser seguido imediatamente por um 1?

36. Quantas cadeias de bits contêm exatamente cinco 0s e quatorze 1s, se todo 0 deve ser seguido imediatamente por dois 1s?

37. Quantas cadeias de bits de extensão 10 contêm pelo menos três 1s e três 0s?

38. Quantas maneiras de selecionar 12 países na ONU para servir em um conselho são possíveis, se 3 são selecionados a partir de um bloco de 45, 4 são selecionados a partir de um bloco de 57 e os outros, a partir dos 69 países restantes?

39. Quantas placas de identificação de carro são possíveis considerando-se que elas são compostas por três letras seguidas de três dígitos, sendo que nenhuma letra ou dígito aparece duas vezes?

40. Há quantas maneiras de seis pessoas sentarem-se em uma mesa circular, onde os assentos são considerados os mesmos se eles podem ser obtidos por uma rotação da mesa?

41. Há quantas maneiras possíveis de terminar uma corrida de cavalos com três cavalos, se podem ocorrer empates? (*Nota:* Dois ou três cavalos podem empatar.)

***42.** Há quantas maneiras possíveis de terminar uma corrida de cavalos com quatro cavalos, se podem ocorrer empates? (*Nota:* Qualquer número de cavalos pode empatar.)

***43.** Há seis atletas em uma corrida de 100 metros. Quantas maneiras de ganhar as três medalhas são possíveis, se podem ocorrer empates? (O atleta ou os atletas que terminarem com o melhor tempo recebem a medalha de ouro, o atleta ou os atletas que terminarem com um corredor à frente recebem medalha de prata e o atleta ou os atletas que terminarem com dois corredores à frente recebem a medalha de bronze.)

***44.** Este procedimento é usado para desempatar os jogos nos campeonatos da Copa do Mundo de Futebol. Cada time escolhe cinco jogadores em uma ordem prescrita. Cada um desses jogadores chuta um pênalti, com um jogador do primeiro time seguido do jogador do segundo time e assim por diante, seguindo a ordem específica dos jogadores. Se o placar continuar empatado no final da primeira rodada de 10 pênaltis, o procedimento é repetido. Se o placar continuar empatado na segunda rodada, um lance de morte-súbita acontece, sendo que o primeiro time que marcar sem ter um gol de resposta vence sem contestações.

a) Quantos placares diferentes são possíveis se o jogo terminar na primeira rodada de 10 pênaltis, sendo que, quando a rodada termina, é impossível para um time igualar o número de gols do outro time?

b) Quantos placares diferentes são possíveis para a primeira e a segunda rodada, se o jogo termina na segunda rodada?

c) Quantos placares são possíveis para um conjunto completo de rodadas de pênaltis, se o jogo termina com não mais de 10 chutes adicionais depois das duas rodadas de cinco chutes para cada time?

5.4 Coeficientes Binomiais

Como vimos na Seção 5.3, o número de r-combinações de um conjunto com n elementos é geralmente representado por $\binom{n}{r}$. Esse número também é chamado de **coeficiente binomial**, porque esses números ocorrem como coeficientes na expansão das potências de expressões binomiais, tais como $(a+b)^n$. Discutiremos o **binômio de Newton**, que nos dá uma potência de uma expressão binomial como uma soma de termos que envolvem coeficientes binomiais. Demonstraremos esse binômio por meio da demonstração combinatorial. Mostraremos também como as demonstrações combinatoriais podem ser usadas para estabelecer algumas das mais diferentes identidades que expressam as relações entre os coeficientes binomiais.

O Binômio de Newton

Links

O binômio de Newton fornece os coeficientes da expansão de potência das expressões binomiais. Uma expressão **binomial** é simplesmente a soma de dois termos, tais como $x + y$. (Os termos podem ser produtos de constantes e variáveis, mas isso não vem ao caso aqui.) O Exemplo 1 mostra por que esse teorema é aplicado.

EXEMPLO 1 A expansão de $(x+y)^3$ pode ser encontrada usando-se a razão combinatória em vez de multiplicar os três termos. Quando $(x+y)^3 = (x+y)(x+y)(x+y)$ é expandido, todos os produtos de um termo na primeira soma, um termo na segunda soma e um termo na terceira soma são adicionados. Termos da forma x^3, x^2y, xy^2 e y^3 aparecem. Para obter um termo na forma x^3, um x deve ser escolhido em cada uma das somas, e isto pode ser feito de apenas uma maneira. Então, o termo x^3 do produto tem um coeficiente 1. Para obter um termo na forma x^2y, um x deve ser escolhido em duas dessas três somas (e, conseqüentemente, um y na outra soma). Assim, o número de tais termos é o número de 2-combinações de três objetos, ou seja, $\binom{3}{2}$. Analogamente, o número de termos na forma xy^2 é o número de maneiras de escolher uma das três somas para obter um x (e, conseqüentemente, um y em cada uma das duas outras somas). Isto pode ser feito de $\binom{3}{1}$ maneiras. Finalmente, a única forma de obter um termo y^3 é escolher o y para cada uma das três somas no produto, e isso pode ser feito de uma maneira. Conseqüentemente, vemos que

$$\begin{aligned}(x+y)^3 &= (x+y)(x+y)(x+y) = (xx+xy+yx+yy)(x+y)\\ &= xxx+xxy+xyx+xyy+yxx+yxy+yyx+yyy\\ &= x^3+3x^2y+3xy^2+y^3.\end{aligned}$$

◀

Podemos agora enunciar o binômio de Newton.

TEOREMA 1 **O BINÔMIO DE NEWTON** Considere x e y como variáveis e n como um número inteiro não negativo. Então,

$$\begin{aligned}(x+y)^n &= \sum_{j=0}^{n}\binom{n}{j}x^{n-j}y^j\\ &= \binom{n}{0}x^n+\binom{n}{1}x^{n-1}y+\binom{n}{2}x^{n-2}y^2+\cdots+\binom{n}{n-1}xy^{n-1}+\binom{n}{n}y^n.\end{aligned}$$

Demonstração: Daremos uma demonstração combinatória desse teorema. Os termos do produto, quando expandido, estão na forma $x^{n-j}y^j$ para $j = 0, 1, 2, \ldots, n$. Para contar o número de termos da forma $x^{n-j}y^j$, é necessário escolher $n-j$ xs a partir de n somas (assim, os outros j termos no produto são ys). Portanto, o coeficiente de $x^{n-j}y^j$ é $\binom{n}{n-j}$, que é igual a $\binom{n}{j}$. Isto demonstra o teorema. ◁

O uso do binômio de Newton está ilustrado nos exemplos 2 a 4.

EXEMPLO 2 Qual é o desenvolvimento de $(x + y)^4$?

Exemplos Extras

Solução: A partir do binômio de Newton, temos que

$$\begin{aligned}(x+y)^4 &= \sum_{j=0}^{4}\binom{4}{j} x^{4-j}y^j \\ &= \binom{4}{0}x^4 + \binom{4}{1}x^3y + \binom{4}{2}x^2y^2 + \binom{4}{3}xy^3 + \binom{4}{4}y^4 \\ &= x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4.\end{aligned}$$

◀

EXEMPLO 3 Qual é o coeficiente de $x^{12}y^{13}$ no desenvolvimento de $(x + y)^{25}$?

Solução: A partir do binômio de Newton, temos que o coeficiente é

$$\binom{25}{13} = \frac{25!}{13!12!} = 5\,200\,300.$$

◀

EXEMPLO 4 Qual é o coeficiente de $x^{12}y^{13}$ no desenvolvimento de $(2x - 3y)^{25}$?

Solução: Primeiro, note que esta expressão é igual a $(2x + (-3y))^{25}$. Pelo binômio de Newton, temos que

$$(2x + (-3y))^{25} = \sum_{j=0}^{25}\binom{25}{j}(2x)^{25-j}(-3y)^j.$$

Conseqüentemente, o coeficiente de $x^{12}y^{13}$ no desenvolvimento é obtido quando $j = 13$, ou seja,

$$\binom{25}{13}2^{12}(-3)^{13} = -\frac{25!}{13!12!}2^{12}3^{13}.$$

◀

Usando o binômio de Newton, podemos demonstrar algumas identidades úteis, como podemos ver nos Corolários 1, 2 e 3.

COROLÁRIO 1 Considere n como um inteiro não negativo. Então,

$$\sum_{k=0}^{n}\binom{n}{k} = 2^n.$$

Demonstração: Usando o binômio de Newton, com $x = 1$ e $y = 1$, vemos que

$$2^n = (1+1)^n = \sum_{k=0}^{n}\binom{n}{k}1^k1^{n-k} = \sum_{k=0}^{n}\binom{n}{k}.$$

Este é o resultado desejado. ◁

Há também uma demonstração combinatorial interessante do Corolário 1, que apresentaremos agora.

Demonstração: Um conjunto com n elementos tem um total de 2^n subconjuntos diferentes. Cada subconjunto tem zero elementos, um elemento, dois elementos . . . ou n elementos. Há $\binom{n}{0}$ subconjuntos com nenhum elemento, $\binom{n}{1}$ subconjuntos com um elemento, $\binom{n}{2}$ subconjuntos com dois elementos . . . e $\binom{n}{n}$ subconjuntos com n elementos. Portanto,

$$\sum_{k=0}^{n} \binom{n}{k}$$

é o número total de subconjuntos de um conjunto com n elementos. Isto mostra que

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n.$$

◁

COROLÁRIO 2 Considere n como um número inteiro positivo. Então,

$$\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0.$$

Demonstração: Quando usamos o binômio de Newton, com $x = -1$ e $y = 1$, vemos que

$$0 = 0^n = ((-1) + 1)^n = \sum_{k=0}^{n} \binom{n}{k} (-1)^k 1^{n-k} = \sum_{k=0}^{n} \binom{n}{k} (-1)^k.$$

Isto demonstra o corolário. ◁

Lembre-se: O Corolário 2 implica que

$$\binom{n}{0} + \binom{n}{2} + \binom{n}{4} + \cdots = \binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \cdots.$$

COROLÁRIO 3 Considere n como um número inteiro não negativo. Então,

$$\sum_{k=0}^{n} 2^k \binom{n}{k} = 3^n.$$

Demonstração: Reconhecemos que o lado esquerdo dessa fórmula é o desenvolvimento de $(1 + 2)^n$ fornecida pelo binômio de Newton. Assim, pelo binômio de Newton, vemos que

$$(1 + 2)^n = \sum_{k=0}^{n} \binom{n}{k} 1^{n-k} 2^k = \sum_{k=0}^{n} \binom{n}{k} 2^k$$

Assim,

$$\sum_{k=0}^{n} 2^k \binom{n}{k} = 3^n.$$

◁

Identidade de Pascal e Triângulo de Pascal

Os coeficientes binomiais satisfazem muitas identidades. Introduziremos agora uma das mais importantes.

TEOREMA 2 **IDENTIDADE DE PASCAL** Considere n e k como números inteiros positivos com $n \geq k$. Então,

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}.$$

Demonstração: Suponha que T seja um conjunto com $n+1$ elementos. Considere a como um elemento de T e considere $S = T - \{a\}$. Note que há $\binom{n+1}{k}$ subconjuntos de T com k elementos. Entretanto, um subconjunto de T com k elementos contém o elemento a juntamente com $k-1$ elementos de S, ou contém k elementos de S e não contém a. Visto que há $\binom{n}{k-1}$ subconjuntos com $k-1$ elementos de S, há $\binom{n}{k-1}$ subconjuntos de k elementos de T que contêm a. E há $\binom{n}{k}$ subconjuntos de k elementos de T que não contêm a, pois há $\binom{n}{k}$ subconjuntos de k elementos de S. Conseqüentemente,

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}.$$

◁

Lembre-se: Mostramos uma demonstração combinatória da identidade de Pascal. Também é possível demonstrar essa identidade pela manipulação algébrica a partir da fórmula para $\binom{n}{r}$ (veja o Exercício 19 no final desta seção).

Lembre-se: A identidade de Pascal, junto com as condições iniciais $\binom{n}{0} = \binom{n}{n} = 1$ para todos os números inteiros n, pode ser usada para definir recursivamente coeficientes binomiais. Essa definição recursiva é útil ao computar coeficientes binomiais, porque apenas a adição, não a multiplicação, de números inteiros é necessária para usar a definição recursiva.

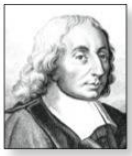
A identidade de Pascal é a base para um arranjo geométrico de coeficientes binomiais em forma de triângulo, como mostra a Figura 1.

A n-ésima linha do triângulo consiste em coeficientes binomiais

$$\binom{n}{k}, k = 0, 1, \ldots, n.$$

Este triângulo é conhecido como **triângulo de Pascal.** A identidade de Pascal mostra que, quando dois coeficientes binomiais adjacentes são somados no triângulo, é produzido um coeficiente binomial na próxima linha entre esses dois coeficientes.

Links



BLAISE PASCAL (1623–1662) Blaise Pascal exibiu seus talentos muito jovem, embora seu pai, que fez descobertas em geometria analítica, tenha mantido os livros de matemática longe dele para incitar-lhe outros interesses. Aos 16 anos, Pascal descobriu um importante resultado em seções cônicas. Aos 18 anos, ele criou uma máquina de calcular, a qual ele mesmo construiu e vendeu. Pascal, junto com Fermat, deixou os fundamentos da teoria moderna da probabilidade. Em seu trabalho, fez novas descobertas, que agora são conhecidas como triângulo de Pascal. Em 1654, Pascal abandonou suas buscas matemáticas para devotar-se à teologia. Depois disso, retornou à matemática apenas uma vez. Uma noite, perturbado por uma severa dor de dente, ele buscou conforto estudando as propriedades matemáticas do cicloide. Milagrosamente, sua dor desapareceu, o que ele sentiu como um sinal de aprovação divina do estudo da matemática.

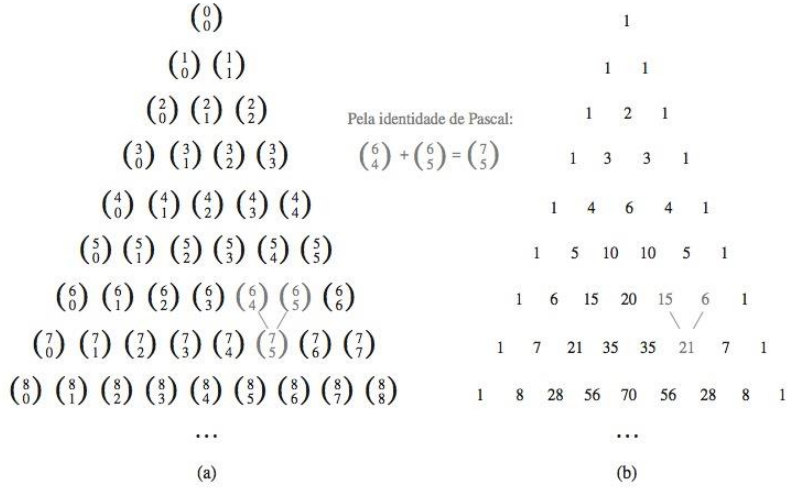


FIGURA 1 **Triângulo de Pascal.**

Algumas Outras Identidades dos Coeficientes Binomiais

Concluímos esta seção com demonstrações combinatoriais de duas das muitas identidades dos coeficientes binomiais.

TEOREMA 3 **IDENTIDADE DE VANDERMONDE** Considere m, n e r como números inteiros não negativos com r não excedente a m ou n. Então,

$$\binom{m+n}{r} = \sum_{k=0}^{r} \binom{m}{r-k}\binom{n}{k}.$$

Lembre-se: Esta identidade foi descoberta pelo matemático Alexandre-Théophile Vandermonde, no século XVIII.

Demonstração: Suponha que haja m itens em um conjunto e n itens em um segundo conjunto. Então, o número total de maneiras de escolher r elementos da união desses conjuntos é $\binom{m+n}{r}$. Outra maneira de escolher r elementos da união é escolher k elementos do primeiro conjunto e, então, escolher $r-k$ elementos do segundo conjunto, em que k é um número inteiro com $0 \leq k \leq r$. Isto pode ser feito de $\binom{m}{k}\binom{n}{r-k}$ maneiras, usando-se a regra do produto. Assim, o número total de maneiras de escolher r elementos da união é também igual a

$$\binom{m+n}{r} = \sum_{k=0}^{r} \binom{m}{r-k}\binom{n}{k}.$$

Isto demonstra a identidade de Vandermonde. ◁

O Corolário 4 diz respeito à identidade de Vandermonde.

COROLÁRIO 4 Se n é um número inteiro não negativo, então,

$$\binom{2n}{n} = \sum_{k=0}^{n} \binom{n}{k}^2$$

Demonstração: Usamos a identidade de Vandermonde com $m = r = n$ e obtemos

$$\binom{2n}{n} = \sum_{k=0}^{n} \binom{n}{n-k}\binom{n}{k} = \sum_{k=0}^{n} \binom{n}{k}^2.$$

A última equação foi obtida usando-se a identidade $\binom{n}{k} = \binom{n}{n-k}$. ◁

Podemos demonstrar as identidades combinatórias pela contagem das cadeias de bits com diferentes propriedades, assim como o Teorema 4 irá demonstrar.

TEOREMA 4 Considere n e r como números inteiros não negativos, com $r \leq n$. Então,

$$\binom{n+1}{r+1} = \sum_{j=r}^{n} \binom{j}{r}.$$

Demonstração: Usamos uma demonstração combinatória. Pelo Exemplo 14 da Seção 5.3, o lado esquerdo da expressão, $\binom{n+1}{r+1}$, conta as cadeias de bits de extensão $n + 1$ que contêm $r + 1$ algarismos 1s.

Mostramos que o lado direito é a contagem dos mesmos objetos, considerando os casos correspondentes às possíveis localizações do último algarismo 1 em uma seqüência com $r + 1$ algarismos 1s. Esse último 1 deve ocorrer na posição $r + 1, r + 2, \ldots$ ou $n + 1$. Além disso, se o último algarismo 1 é o k-ésimo bit, deve haver r algarismos 1s entre as primeiras $k - 1$ posições. Conseqüentemente, pelo Exemplo 14 da Seção 5.3, há $\binom{k-1}{r}$ de tais cadeias de bits. Somados sobre k com $r + 1 \leq k \leq n + 1$, encontramos

$$\sum_{k=r+1}^{n+1} \binom{k-1}{r} = \sum_{j=r}^{n} \binom{j}{r}$$

cadeias de bits de extensão n com exatamente $r + 1$ algarismos 1s. (Note que o último passo decorre da mudança de variáveis $j = k - 1$.) Como o lado esquerdo e o lado direito da equação fazem a contagem dos mesmos objetos, eles são iguais. Isto completa a demonstração. ◁

Links

ALEXANDRE-THÉOPHILE VANDERMONDE (1735–1796) Pelo fato de Alexandre-Théophile Vandermonde ter sido uma criança doente, seu pai, médico, direcionou-o para a carreira musical. Entretanto, ele acabou por desenvolver um interesse pela matemática. Seu trabalho completo em matemática consiste em quatro artigos publicados entre 1771 e 1772. Esses artigos incluem contribuições fundamentais em raízes de equações, na teoria de determinantes e no problema do passeio do cavalo (introduzido nos exercícios da Seção 9.5). O interesse de Vandermonde por matemática durou apenas 2 anos. Depois disso, ele publicou artigos sobre harmonia, experimentos com resfriamentos e manufatura de aço. Ele também tinha interesse por política, juntando-se à causa da Revolução Francesa e ocupando diversos cargos no governo.

Exercícios

1. Encontre o desenvolvimento de $(x+y)^4$
   a) usando o raciocínio combinatório, como no Exemplo 1.
   b) usando o binômio de Newton.
2. Encontre o desenvolvimento de $(x+y)^5$
   a) usando o raciocínio combinatório, como no Exemplo 1.
   b) usando o binômio de Newton.
3. Encontre o desenvolvimento de $(x+y)^6$.
4. Encontre o coeficiente de x^5y^8 em $(x+y)^{13}$.
5. Quantos termos haverá no desenvolvimento de $(x+y)^{100}$, depois que os termos semelhantes forem agrupados?
6. Qual o coeficiente de x^7 em $(1+x)^{11}$?
7. Qual o coeficiente de x^9 em $(2-x)^{19}$?
8. Qual é o coeficiente de $x^8\,y^9$ no desenvolvimento de $(3x+2y)^{17}$?
9. Qual é o coeficiente de $x^{101}\,y^{99}$ no desenvolvimento de $(2x-3y)^{200}$?
10. Dê a fórmula para o coeficiente de x^k no desenvolvimento de $(x+1/x)^{100}$, em que k é um número inteiro.
11. Dê a fórmula para o coeficiente de x^k no desenvolvimento de $(x^2-1/x)^{100}$, em que k é um número inteiro.
12. A linha do triângulo de Pascal que contém os coeficientes binomiais $\binom{10}{k}$, $0 \le k \le 10$, é:

    1 10 45 120 210 252 210 120 45 10 1

    Use a identidade de Pascal para produzir a linha imediatamente subseqüente a esta linha no triângulo de Pascal.
13. Qual é a linha do triângulo de Pascal que contém o coeficiente binomial $\binom{9}{k}$, $0 \le k \le 9$?
14. Mostre que se n é um número inteiro positivo, então

    $$1 = \binom{n}{0} < \binom{n}{1} < \cdots < \binom{n}{\lfloor n/2 \rfloor} = \binom{n}{\lfloor n/2 \rfloor} > \cdots > \binom{n}{n-1} > \binom{n}{n} = 1.$$

15. Mostre que $\binom{n}{k} \le 2^n$ para todo inteiro positivo n e todo inteiro k com $0 \le k \le n$.
16. **a)** Use o Exercício 14 e o Corolário 1 para mostrar que se n for um número inteiro maior que 1, então $\binom{n}{\lfloor n/2 \rfloor} \ge 2^n/n$.
    b) Conclua a partir do item (a) que se n é um número inteiro positivo, então $\binom{2n}{n} \ge 4^n/2n$.
17. Mostre que se n e k são números inteiros com $1 \le k \le n$, então $\binom{n}{k} \le n^k/2^{k-1}$.
18. Suponha que b seja um número inteiro com $b \ge 7$. Use o binômio de Newton e a linha apropriada do triângulo de Pascal para encontrar a expansão de base b de $(11)_b^4$ [ou seja, a quarta potência do número $(11)_b$ na base b].
19. Demonstre a identidade de Pascal usando a fórmula para $\binom{n}{r}$.
20. Suponha que k e n sejam números inteiros com $1 \le k < n$. Demonstre a **identidade do hexágono**

    $$\binom{n-1}{k-1}\binom{n}{k+1}\binom{n+1}{k} = \binom{n-1}{k}\binom{n}{k-1}\binom{n+1}{k+1},$$

    que relaciona os termos do triângulo de Pascal que formam um hexágono.
21. Demonstre que se n e k forem números inteiros com $1 \le k \le n$, então $k\binom{n}{k} = n\binom{n-1}{k-1}$
    a) usando uma demonstração combinatória. [*Dica:* Mostre que os dois lados da identidade fazem a contagem do número de maneiras de selecionar um subconjunto com k elementos a partir de um conjunto com n elementos e então um elemento desse subconjunto.]
    b) usando uma demonstração algébrica baseada na fórmula para $\binom{n}{r}$, dada no Teorema 2 da Seção 5.3.
22. Demonstre a identidade $\binom{n}{r}\binom{r}{k} = \binom{n}{k}\binom{n-k}{r-k}$, sempre que n, r e k forem números inteiros não negativos com $r \le n$ e $k \le r$,
    a) usando um argumento combinatório.
    b) usando um argumento baseado na fórmula para o número de r-combinações de um conjunto com n elementos.
23. Mostre que se n e k forem números inteiros positivos, então

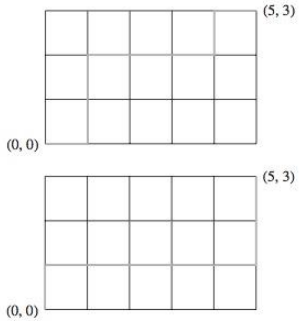
    $$\binom{n+1}{k} = (n+1)\binom{n}{k-1}/k.$$

    Use essa identidade para construir uma definição indutiva dos coeficientes binomiais.
24. Mostre que se p for um número primo e k um número inteiro, tal que $1 \le k \le p-1$, então p divide $\binom{p}{k}$.
25. Considere n como um número inteiro positivo. Mostre que $\binom{2n}{n+1} + \binom{2n}{n} = \binom{2n+2}{n+1}/2$.
26. Considere n e k como números inteiros com $1 \le k \le n$. Mostre que $\sum_{k=1}^{n}\binom{n}{k}\binom{n}{k-1} = \binom{2n+2}{n+1}/2 - \binom{2n}{n}$.
27. Demonstre que

    $$\sum_{k=0}^{r}\binom{n+k}{k} = \binom{n+r+1}{r}$$

    sempre que n e r forem números inteiros positivos,
    a) usando um argumento combinatório.
    b) usando a identidade de Pascal.
28. Mostre que se n for um número inteiro positivo, então $\binom{2n}{2} = 2\binom{n}{2} + n^2$
    a) usando um argumento combinatório.
    b) pela manipulação algébrica.
29. Dê uma demonstração combinatorial de $\sum_{k=1}^{n} k\binom{n}{k} = n2^{n-1}$. [*Dica:* Conte de duas formas o número de maneiras de escolher um comitê e então selecione um líder do comitê.]
30. Dê uma demonstração combinatorial de $\sum_{k=1}^{n} k\binom{n}{k}^2 = n\binom{2n-1}{n-1}$. [*Dica:* Conte de duas formas o número de maneiras de selecionar um comitê com n membros de um grupo de n professores de matemática e n professores de ciência da computação, tal que o presidente do comitê seja um professor de matemática.]
31. Mostre que um conjunto não vazio tem o mesmo número de subconjuntos com um número ímpar de elementos que subconjuntos com número par de elementos.
32. Demonstre o binômio de Newton usando a indução matemática.
33. Neste exercício contaremos o número de caminhos no plano xy entre a origem $(0, 0)$ e o ponto (m, n), de tal modo que

cada caminho seja construído a partir de uma série de passos, em que cada passo seja um movimento de uma unidade para a direita ou um movimento de uma unidade para cima. (Movimentos para a esquerda ou para baixo não são permitidos.) Dois caminhos de (0, 0) a (5, 3) são ilustrados aqui.



a) Mostre que cada caminho do tipo descrito acima pode ser representado por uma cadeia de bits que contenha m 0s e n 1s, em que um 0 representa um movimento de uma unidade para a direita e 1 representa um movimento de uma unidade para cima.

b) Conclua a partir do item (a) que há $\binom{m+n}{n}$ caminhos do tipo desejado.

34. Use o Exercício 33 para demonstrar que $\binom{n}{k} = \binom{n}{n-k}$, sempre que k for um número inteiro com $0 \leq k \leq n$. [*Dica:* Considere o número de caminhos do tipo descrito no Exercício 33 de (0, 0) a $(n-k, k)$ e de (0, 0) a $(k, n-k)$.]

35. Use o Exercício 33 para demonstrar o Teorema 4. [*Dica:* Conte o número de caminhos com n passos do tipo descrito no Exercício 33. Todo caminho deve terminar com um dos pontos $(n-k, k)$ para $k = 0, 1, 2, \ldots, n$.]

36. Use o Exercício 33 para demonstrar a identidade de Pascal. [*Dica:* Mostre que um caminho do tipo descrito no Exercício 33 de (0, 0) a $(n+1-k, k)$ passa por $(n+1-k, k-1)$ ou por $(n-k, k)$, mas não por ambos.]

37. Demonstre a identidade no Exercício 27 usando o Exercício 33. [*Dica:* Primeiro, note que o número de caminhos de (0, 0) a $(n+1, r)$ é igual a $\binom{n+1+r}{r}$. Segundo, conte o número de caminhos pela soma dos números desses caminhos que começam com k unidades para cima, para $k = 0, 1, 2, \ldots, r$.]

38. Dê uma demonstração combinatorial de que se n for um número inteiro positivo, então $\Sigma_{k=0}^{n} k^2 \binom{n}{k} = n(n+1)2^{n-2}$. [*Dica:* Mostre que os dois lados fazem a contagem das maneiras de escolher um subconjunto de um conjunto com n elementos junto com dois elementos não necessariamente distintos deste subconjunto. Além disso, expresse o lado direito como $n(n-1)2^{n-2} + n2^{n-1}$.]

***39.** Determine uma fórmula que envolva coeficientes binomiais para o n-ésimo termo de uma seqüência se seus termos iniciais forem os listados. [*Dica:* Trabalhar com o triângulo de Pascal será útil. Embora sejam infinitas, é possível perceber que cada uma das listas abaixo é o início de uma seqüência do tipo desejado.]

a) 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, . . .

b) 1, 4, 10, 20, 35, 56, 84, 120, 165, 220, . . .

c) 1, 2, 6, 20, 70, 252, 924, 3432, 12870, 48620, . . .

d) 1, 1, 2, 3, 6, 10, 20, 35, 70, 126, . . .

e) 1, 1, 1, 3, 1, 5, 15, 35, 1, 9, . . .

f) 1, 3, 15, 84, 495, 3003, 18564, 116280, 735471, 4686825, . . .

5.5 Permutações e Combinações Generalizadas

Introdução

Em muitos problemas de contagem, os elementos podem ser usados repetidamente. Por exemplo, uma letra ou um dígito podem ser usados mais de uma vez em uma placa de identificação de um automóvel. Quando uma dúzia de rosquinhas são escolhidas, cada tipo pode ser escolhido repetidamente. Isto contrasta com os problemas de contagem discutidos anteriormente neste capítulo, em que consideramos apenas permutações e combinações nas quais cada item podia ser usado quando muito uma vez. Nesta seção, mostraremos como resolver problemas de contagem em que os elementos podem ser usados mais de uma vez.

Além disso, alguns problemas de contagem envolvem elementos idênticos. Por exemplo, para contar o número de maneiras que as letras da palavra *SUCESS* podem ser rearranjadas, o lugar das letras idênticas deve ser considerado. Isto contrasta com os problemas de contagem discutidos anteriormente, em que todos os elementos eram considerados idênticos. Nesta seção, descreveremos como resolver os problemas de contagem nos quais alguns elementos são idênticos.

Explicaremos, também, como resolver outra importante classe de problemas de contagem que envolve a contagem das maneiras de colocar elementos distintos em caixas. Um exemplo desse tipo de problemas é o número de maneiras diferentes de mãos de pôquer com quatro jogadores.

Utilizados concomitantemente, os métodos descritos anteriormente neste capítulo e os métodos introduzidos nesta seção formam uma ferramenta muito útil para a resolução de uma grande variedade de problemas de contagem. Quando os métodos adicionais, discutidos no Capítulo 7, forem somados a estes, você será capaz de resolver inúmeros problemas de contagem que aparecem em muitas áreas de estudo.

Permutações com Repetição

As permutações com repetição podem ser facilmente realizadas com o uso da regra do produto, como mostra o Exemplo 1.

EXEMPLO 1 Quantas seqüências de extensão r podem ser formadas com o alfabeto da língua inglesa?

Solução: Usando a regra do produto temos: como há 26 letras e como cada letra pode ser usada repetidamente, vemos que são possíveis 26^r seqüências de extensão r. ◀

O número de r-permutações de um conjunto com n elementos, quando há repetição, pode ser resolvido com o Teorema 1.

TEOREMA 1 O número de r-permutações de um conjunto com n objetos, com repetição, é n^r.

Demonstração: Há n maneiras possíveis de escolher um elemento do conjunto para cada uma das r posições na r-permutação, com repetição, porque para cada escolha todos os n objetos estão disponíveis. Assim, pela regra do produto, são possíveis n^r r-permutações com repetição. ◁

Combinações com Repetição

Considere os exemplos abaixo de combinação com repetição de elementos.

EXEMPLO 2 Há quantas maneiras possíveis de escolher quatro pedaços de fruta a partir de uma tigela que contém maçãs, laranjas e pêras, se a ordem em que cada pedaço é escolhido não é relevante, apenas o tipo de fruta, e o pedaço individual não é relevante também, sendo que há pelo menos quatro pedaços de cada tipo de fruta na tigela?

Solução: Para resolver esse problema, listamos todas as maneiras possíveis de escolher uma fruta. Há 15 maneiras:

4 maçãs	4 laranjas	4 pêras
3 maçãs, 1 laranja	3 maçãs, 1 pêra	3 laranjas, 1 maçã
3 laranjas, 1 pêra	3 pêras, 1 maçã	3 pêras, 1 laranja
2 maçãs, 2 laranjas	2 maçãs , 2 pêras	2 laranjas, 2 pêras
2 maçãs, 1 laranja, 1 pêra	2 laranjas, 1 maçã, 1 pêra	2 pêras, 1 maçã, 1 laranja

A solução é o número de 4-combinações com repetição de um conjunto de três elementos, {*maçã*, *laranja*, *pêra*}. ◀

Para resolver problemas de contagem mais complexos desse tipo, necessitamos de um método geral para a contagem de r-combinações de um conjunto com n elementos. No Exemplo 3, mostraremos tal método.

FIGURA 1 **Caixa de Dinheiro com Sete Tipos de Cédulas.**

EXEMPLO 3 Há quantas maneiras de escolher cinco cédulas possíveis em uma caixa que contém cédulas de R$ 1, R$ 2, R$ 5, R$ 10, R$ 20, R$ 50 e R$ 100? Considere que a ordem na qual cada cédula é escolhida não é relevante, que cada nota é idêntica e que há pelo menos cinco cédulas de cada valor.

Solução: Como a ordem em que as cédulas são escolhidas não é relevante e sete tipos diferentes de cédula podem ser escolhidas cinco vezes, este problema envolve a contagem de 5-combinações com repetição de um conjunto com sete elementos. Listar todas as possibilidades pode se tornar demorado e tedioso, pois há um grande número de soluções. Em vez disso, mostraremos o uso de uma técnica de contagem de combinações com repetição.

Suponha que uma caixa tenha sete compartimentos, uma para cada tipo de cédula, como ilustrado na Figura 1. Esses compartimentos são separados por seis divisórias, como mostrado na figura. A escolha de cinco cédulas corresponde a colocar cinco marcadores nos compartimentos para cédulas diferentes. A Figura 2 ilustra esta correspondência de três maneiras diferentes de escolher as cinco cédulas, em que os seis divisores são representados por barras e as cinco cédulas por asteriscos.

O número de maneiras de escolher as cinco cédulas corresponde ao número de maneiras de organizar seis barras e cinco asteriscos. Conseqüentemente, o número de maneiras de escolher cinco cédulas é o número de maneiras de escolher as posições dos cinco asteriscos, a partir de 11 posições possíveis. Isto corresponde ao número de seleções não ordenadas de 5 objetos de um conjunto de 11 objetos, o que pode se feito de $C(11, 5)$ maneiras. Conseqüentemente, há

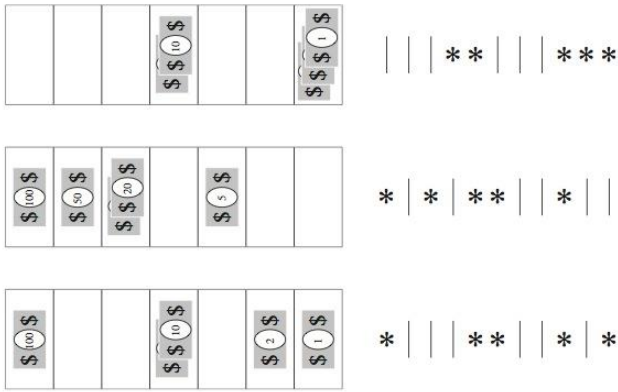


FIGURA 2 **Exemplos de Maneiras de Escolher Cinco Cédulas.**

$$C(11, 5) = \frac{11!}{5!6!} = 462$$

maneiras possíveis de escolher cinco cédulas em uma caixa com sete tipos de cédulas. ◀

O Teorema 2 generaliza esta discussão.

TEOREMA 2 Há $C(n + r - 1, r) = C(n + r - 1, n - 1)$ r-combinações de um conjunto com n elementos, quando a repetição dos elementos é permitida.

Demonstração: Cada uma das r-combinações de um conjunto com n elementos, quando a repetição é permitida, pode ser representada por uma lista de $n - 1$ barras e r asteriscos. Então, $n - 1$ barras são usadas para representar n células diferentes, com a i-ésima célula contendo um asterisco para toda vez que o i-ésimo elemento do conjunto aparecer na combinação. Por exemplo, uma 6-combinação de um conjunto com quatro elementos é representada com três barras e seis asteriscos. Aqui,

$$**|*||***$$

representa a combinação que contém exatamente duas vezes o primeiro elemento, uma o segundo elemento, nenhuma o terceiro elemento e três vezes o quarto elemento do conjunto.

Como vimos, cada lista diferente contém $n - 1$ barras e r asteriscos, que correspondem a uma r-combinação do conjunto com n elementos, com repetição. O número de tais listas é $C(n - 1 + r, r)$, pois cada lista corresponde a uma escolha de r posições para colocar os r asteriscos a partir de $n - 1 + r$ posições que contêm r asteriscos e $n - 1$ barras. O número de tais listas é também igual a $C(n - 1 + r, n - 1)$, pois cada lista corresponde a uma escolha de $n - 1$ posições para colocar as $n - 1$ barras. ◁

Os exemplos 4 a 6 mostram como o Teorema 2 é aplicado.

EXEMPLO 4 Suponha que uma cafeteria tenha quatro tipos diferentes de biscoitos. Há quantas maneiras diferentes possíveis de escolher seis biscoitos? Suponha que apenas o tipo de biscoito seja relevante, e não os biscoitos individualmente ou a ordem em que são escolhidos.

Exemplos Extras

Solução: O número de maneiras de escolher seis biscoitos é o número de 6-combinações de um conjunto com quatro elementos. Pelo Teorema 2, isto é igual a $C(4 + 6 - 1, 6) = C(9, 6)$. Como

$$C(9, 6) = C(9, 3) = \frac{9 \cdot 8 \cdot 7}{1 \cdot 2 \cdot 3} = 84,$$

há 84 maneiras diferentes possíveis de escolher seis biscoitos. ◀

O Teorema 2 também pode ser usado para encontrar o número de soluções para certas equações lineares, em que as variáveis são números inteiros sujeitos a uma restrição. Isto será mostrado no Exemplo 5.

EXEMPLO 5 Quantas soluções a equação

$$x_1 + x_2 + x_3 = 11$$

pode ter, em que x_1, x_2 e x_3 são números inteiros não negativos?

Solução: Para contar o número de soluções, notamos que uma solução corresponde a uma maneira de selecionar 11 itens de um conjunto com três elementos, tal que x_1 itens do tipo um, x_2 itens

do tipo dois e x_3 itens do tipo três sejam escolhidos. Assim, o número de soluções é igual ao número de 11-combinações, com repetição, de um conjunto com três elementos. Pelo Teorema 2, verificamos que há

$$C(3+11-1,11) = C(13,11) = C(13,2) = \frac{13 \cdot 12}{1 \cdot 2} = 78$$

soluções.

O número de soluções desta equação pode também ser encontrado quando as variáveis estão sujeitas a restrição. Por exemplo, podemos encontrar o número de soluções em que as variáveis são números inteiros com $x_1 \geq 1$, $x_2 \geq 2$ e $x_3 \geq 3$. Uma solução dessa equação sujeita a essas restrições corresponde a uma seleção de 11 itens, com x_1 itens do tipo um, x_2 itens do tipo dois e x_3 itens do tipo três, em que há pelo menos um item do tipo um, dois itens do tipo dois e três itens do tipo três. Então, escolhemos um item do tipo um, dois do tipo dois e três do tipo três. Depois, selecionamos cinco itens adicionais. Pelo Teorema 2, isto pode ser feito de

$$C(3+5-1,5) = C(7,5) = C(7,2) = \frac{7 \cdot 6}{1 \cdot 2} = 21$$

maneiras. Portanto, há 21 soluções possíveis para a equação sujeita às restrições dadas. ◀

O Exemplo 6 mostra como contar o número de combinações com repetição que aparecem para determinar o valor de uma variável que está sendo incrementada cada vez que um certo tipo de laços agrupados é dado.

EXEMPLO 6 Qual o valor de k depois que o pseudocódigo abaixo for executado?

```
k := 0
for i_1 := 1 to n
  for i_2 := 1 to i_1
        .
        .
        .
    for i_m := 1 to i_{m-1}
        k := k + 1
```

Solução: Note que o valor inicial de k é 0 e que 1 é adicionado a k toda vez que o laço for dado com uma seqüência de inteiros $i_1, i_2, \ldots, i_m$, tal que

$$1 \leq i_m \leq i_{m-1} \leq \cdots \leq i_1 \leq n.$$

O número de tais seqüências é o número de maneiras de escolher m números inteiros de $\{1, 2, \ldots, n\}$, com repetição. (Para verificar isso, note que uma vez que a seqüência for escolhida, se a ordem dos números inteiros for uma ordem não decrescente, esta unicidade define uma tarefa de i_m, $i_{m-1}, \ldots, i_1$. Contrariamente, cada tarefa corresponde a um único conjunto não ordenado.) Assim, pelo Teorema 2, temos que $k = C(n+m-1, m)$ depois que o código tiver sido executado. ◀

As fórmulas para os números de seleções ordenadas e não ordenadas de r elementos, com ou sem repetições, de um conjunto com n elementos, são apresentadas na Tabela 1.

TABELA 1 Combinações e Permutações com e sem Repetição.

Tipo	*Com repetição?*	*Fórmula*
r-permutações	Não	$\frac{n!}{(n-r)!}$
r-combinações	Não	$\frac{n!}{r!(n-r)!}$
r-permutações	Sim	n^r
r-combinações	Sim	$\frac{(n+r-1)!}{r!(n-1)!}$

Permutações com Objetos Idênticos

Alguns elementos podem ser indistinguíveis em problemas de contagem. Quando este é o caso, alguns cuidados devem ser tomados para evitar a contagem de objetos mais de uma vez. Considere o Exemplo 7.

EXEMPLO 7 Quantas seqüências diferentes podem ser obtidas reorganizando-se as letras da palavra

Exemplos Extras

SUCCESS?

Solução: Como algumas das letras de *SUCCESS* são idênticas, *não* será considerado o número de permutações de sete letras. Essa palavra contém três *S*s, dois *C*s, um *U* e um *E*. Para determinar o número de seqüências diferentes que podem ser obtidas reorganizando-se as letras, primeiro note que os três *S*s podem ser colocados entre as sete posições de $C(7, 3)$ maneiras diferentes, deixando quatro posições livres. Então, os dois *C*s podem ser colocados de $C(4, 2)$ maneiras, deixando três posições livres. O *U* pode ser colocado de $C(2, 1)$ maneiras, deixando apenas uma posição livre. Assim, o *E* pode ser colocado de $C(1, 1)$ maneira. Conseqüentemente, pela regra do produto, o número de seqüências diferentes que podem ser feitas é

$$\begin{aligned} C(7,3)C(4,2)C(2,1)C(1,1) &= \frac{7!}{3!4!} \cdot \frac{4!}{2!2!} \cdot \frac{2!}{1!1!} \cdot \frac{1!}{1!0!} \\ &= \frac{7!}{3!\,2!\,1!\,1!} \\ &= 420. \end{aligned}$$

◀

Podemos demonstrar o Teorema 3 usando o mesmo tipo de raciocínio do Exemplo 7.

TEOREMA 3 O número de permutações diferentes de n objetos, em que há n_1 objetos idênticos do tipo 1, n_2 objetos idênticos do tipo 2, . . . e n_k objetos idênticos do tipo k, é

$$\frac{n!}{n_1!\,n_2!\ldots n_k!}.$$

Demonstração: Para determinar o número de permutações, primeiro note que os n_1 objetos do tipo 1 podem ser colocados em n posições de $C(n, n_1)$ maneiras, deixando $n - n_1$ posições livres. Então, os objetos do tipo 2 podem ser colocados de $C(n - n_1, n_2)$ maneiras, deixando

$n - n_1 - n_2$ posições livres. Continue a colocar os objetos do tipo 3,. . . , tipo $k - 1$, até a última etapa, em que n_k objetos do tipo k podem ser colocados de $C(n - n_1 - n_2 - \cdots - n_{k-1}, n_k)$ maneiras. Assim, pela regra do produto, o número total de permutações diferentes é

$$\begin{aligned} &C(n, n_1)C(n - n_1, n_2)\cdots C(n - n_1 - \cdots - n_{k-1}, n_k) \\ &\quad = \frac{n!}{n_1!(n - n_1)!}\frac{(n - n_1)!}{n_2!(n - n_1 - n_2)!}\cdots\frac{(n - n_1 - \cdots - n_{k-1})!}{n_k!\,0!} \\ &\quad = \frac{n!}{n_1!\,n_2!\cdots n_k!}. \end{aligned}$$

◁

Distribuição de Objetos em Caixas

Links

Muitos problemas de contagem podem ser resolvidos enumerando-se as maneiras como os objetos podem ser colocados em caixas (em que a ordem desses objetos não é relevante). Os objetos podem ser *distintos*, ou seja, diferentes uns dos outros, ou *idênticos*. Os objetos distintos são às vezes chamados de *rotulados*, enquanto os objetos idênticos são chamados de *não rotulados*. Do mesmo modo, as caixas podem ser distintas, ou seja, diferentes, ou idênticas. Caixas distintas são geralmente chamadas de *rotuladas*, enquanto as caixas idênticas, de *não rotuladas*. Quando você resolve um problema de contagem usando o modelo de distribuição de objetos em caixas, precisa determinar se os objetos são distintos e se as caixas são distintas. Embora o contexto do problema de contagem esclareça este ponto, esse problemas são muitas vezes ambíguos e isto pode não deixar claro qual modelo aplicar. Neste caso, o melhor é estabelecer quais hipóteses você está fazendo e explicar o porquê da escolha de determinado modelo, conforme suas hipóteses.

Exemplos Extras

Veremos que há fórmulas fechadas para contar as maneiras de distribuir objetos, distintos ou idênticos, em caixas distintas. Não teremos tanta sorte quando o problema for contar as maneiras de distribuir os objetos, distintos ou idênticos, em caixas idênticas; não há fórmulas fechadas para usar nesses casos.

OBJETOS DISTINTOS E CAIXAS DISTINTAS Primeiro consideramos o caso quando objetos distintos são colocados em caixas distintas. Considere o Exemplo 8, no qual os objetos são cartas e as caixas são as mãos de jogadores.

EXEMPLO 8 Quantas maneiras existem de distribuir mãos de 5 cartas para cada um dos quatro jogadores, a partir de um baralho de 52 cartas?

Solução: Usaremos a regra do produto para resolver este problema. Para começar, note que o primeiro jogador pode ter 5 cartas de $C(52, 5)$ maneiras. O segundo jogador pode ter 5 cartas de $C(47, 5)$ maneiras, porque sobraram apenas 47 cartas. O terceiro jogador pode receber 5 cartas de $C(42, 5)$ maneiras. Finalmente, o quarto jogador pode ter 5 cartas de $C(37, 5)$ maneiras. Assim, o número total de maneiras de dar as cartas aos quatro jogadores, com cinco cartas cada, é

$$\begin{aligned} C(52, 5)C(47, 5)C(42, 5)C(37, 5) &= \frac{52!}{47!5!}\cdot\frac{47!}{42!5!}\cdot\frac{42!}{37!5!}\cdot\frac{37!}{32!5!} \\ &= \frac{52!}{5!5!5!5!32!} \end{aligned}$$

◀

Lembre-se: A solução do Exemplo 8 é igual ao número de permutações de 52 objetos, com 5 objetos distintos de quatro tipos diferentes e 32 objetos de um quinto tipo. Esta equação pode ser

verificada definindo-se uma correspondência um para um, uma função bijetora, entre permutações deste tipo e distribuição de cartas aos jogadores. Para definir essa correspondência, primeiro ordene as cartas de 1 a 52. Então, as cartas dadas ao primeiro jogador correspondem às cartas nas posições dos objetos do primeiro tipo na permutação. Do mesmo modo, as cartas dadas ao segundo, terceiro e quarto jogadores, respectivamente, correspondem às cartas nas posições dos objetos do segundo, terceiro e quarto tipo, respectivamente. As cartas não distribuídas correspondem às cartas na posição dos objetos do quinto tipo. O leitor deverá verificar que esta é uma correspondência um para um.

O Exemplo 8 é um problema típico que envolve distribuição de objetos distintos em caixas distintas. Os objetos distintos são as 52 cartas, e as 5 caixas distintas são as mãos dos quatro jogadores e o resto do baralho. Problemas de contagem que envolvam distribuição de objetos distintos em caixas podem ser resolvidos usando-se o Teorema 4.

TEOREMA 4 O número de maneiras de distribuir n objetos distintos em k caixas distintas, para que os n_i objetos sejam colocados na caixa i, $i = 1, 2, \ldots, k$, é igual a

$$\frac{n!}{n_1!\,n_2!\cdots n_k!}.$$

O Teorema 4 pode ser demonstrado usando-se a regra do produto. Estudaremos os detalhes no Exercício 47. Ele também pode ser demonstrado pela correspondência um para um (veja o Exercício 48) entre as permutações contadas pelo Teorema 3 e as maneiras de distribuir objetos contados pelo Teorema 4.

OBJETOS IDÊNTICOS E CAIXAS DISTINTAS Contar o número de maneiras de colocar n objetos idênticos em k caixas distintas é o mesmo que contar o número de n-combinações de um conjunto com k elementos, com repetição. A razão disto é que há correspondência um para um entre as n-combinações de um conjunto com k elementos, com repetição, e as maneiras de colocar n bolas idênticas em k caixas distintas. Para construir tal correspondência, colocamos uma bola na i-ésima caixa cada vez que o i-ésimo elemento do conjunto estiver incluso na n-combinação.

EXEMPLO 9 Há quantas maneiras possíveis de colocar 10 bolas idênticas em oito caixas distintas?

Solução: O número de maneiras de colocar 10 bolas idênticas em oito caixas é igual ao número de 10-combinações de um conjunto com oito elementos, com repetição. Conseqüentemente, há

$$C(8 + 10 - 1,\ 10) = C(17, 10) = \frac{17!}{10!7!} = 19\,448.$$

◀

Isto significa que há $C(n + r - 1, n - 1)$ maneiras de colocar r objetos idênticos em n caixas distintas.

OBJETOS DISTINTOS E CAIXAS IDÊNTICAS Contar as maneiras de colocar n objetos distintos em k caixas idênticas é mais difícil que contar as maneiras de colocar objetos, distintos ou idênticos, em caixas distintas. Mostraremos isto com um exemplo.

EXEMPLO 10 Há quantas maneiras possíveis de colocar quatro empregados diferentes em três escritórios idênticos, em que cada escritório pode conter qualquer número de empregados?

Solução: Resolveremos este problema enumerando as maneiras possíveis de os empregados serem colocados nos escritórios. Representamos os quatro empregados por A, B, C e D. Primeiro, notamos que podemos distribuir os empregados para que todos fiquem em uma sala, ou três fiquem em uma sala e um em uma segunda sala, ou ainda dois empregados em cada sala e uma sala vazia e, por fim, dois fiquem em uma sala e cada um dos outros dois em uma das outras duas salas. Cada maneira de distribuir os empregados nessas salas representa um maneira de repartir os elementos A, B, C e D em subconjuntos disjuntos.

Podemos colocar os quatro empregados em uma sala de uma maneira, representada por $\{\{A, B, C, D\}\}$. Podemos colocar três empregados em uma sala e o quarto em outra de quatro maneiras, representadas por $\{\{A, B, C\}, \{D\}\}$, $\{\{A, B, D\}, \{C\}\}$, $\{\{A, C, D\}, \{B\}\}$ e $\{\{B, C, D\}, \{A\}\}$. Podemos colocar dois empregados em cada sala de três maneiras, representadas por $\{\{A, B\}, \{C, D\}\}$, $\{\{A, C\}, \{B, D\}\}$ e $\{\{A, D\}, \{B,C\}\}$. Por fim, podemos colocar dois empregados em uma sala e cada um dos outros dois em salas diferentes de seis maneiras, representadas por $\{\{A, B\}, \{C\}, \{D\}\}$, $\{\{A, C\}, \{B\}, \{D\}\}$, $\{\{A, D\}, \{B\}, \{C\}\}$, $\{\{B, C\}, \{A\}, \{D\}\}$, $\{\{B, D\}\}, \{A\}, \{C\}\}$ e $\{\{C, D\}, \{A\}, \{B\}\}$.

Contando todas as possibilidades, encontramos 14 maneiras de colocar quatro empregados diferentes em três salas iguais. Outra forma de olhar para este problema é ver o número de salas nas quais se vão colocar os empregados. Note que há seis maneiras de colocar quatro empregados diferentes em três salas idênticas para que nenhuma sala fique vazia, sete maneiras de colocar os quatro empregados em duas salas iguais, para que nenhuma sala fique vazia, e uma maneira de colocar os quatro empregados em um única sala. ◀

Não há uma fórmula fechada simples para o número de maneiras de distribuir n objetos distintos em j caixas idênticas. Entretanto, há uma fórmula particularmente complicada que descreveremos agora; os leitores podem omitir a discussão desta fórmula, se desejarem. Considere $S(n, j)$ como o número de maneiras de distribuir n objetos distintos em j caixas idênticas, para que nenhuma caixa fique vazia. Os números $S(n, j)$ são chamados de **números de Stirling de segunda ordem.** Por exemplo, o Exemplo 10 mostra que $S(4, 3) = 6$, $S(4, 2) = 7$ e $S(4, 1) = 1$. Vemos que o número de maneiras de distribuir n objetos distintos em k caixas idênticas (em que o número de caixas não vazias é igual a $k, k - 1, \ldots, 2$ ou 1) é igual a $\Sigma_{j=1}^{k} S(n, j)$. Por exemplo, seguindo o raciocínio do Exemplo 10, o número de maneiras de distribuir quatro objetos distintos em três caixas idênticas é igual a $S(4, 1) + S(4, 2) + S(4, 3) = 1 + 7 + 6 = 14$. Usando o princípio da inclusão–exclusão (veja a Seção 7.6), podemos mostrar que

$$S(n, j) = \frac{1}{j!} \sum_{i=0}^{j-1} (-1)^i \binom{j}{i} (j - i)^n.$$

Conseqüentemente, o número de maneiras de distribuir n objetos distintos em k caixas idênticas é igual a

$$\sum_{j=1}^{k} S(n, j) = \sum_{j=1}^{k} \frac{1}{j!} \sum_{i=0}^{j-1} (-1)^i \binom{j}{i} (j - i)^n.$$

(Para mais informações sobre os números de Stirling de segunda ordem, veja os livros teóricos sobre análise combinatória, como [Bó07], [Br99] e [RoTe05] e o Capítulo 6 em [MiRo91].)

OBJETOS IDÊNTICOS E CAIXAS IDÊNTICAS Alguns problemas de contagem podem ser resolvidos pela determinação do número de maneiras de distribuir objetos idênticos em caixas idênticas. Mostraremos este princípio com um exemplo.

EXEMPLO 11 Há quantas maneiras possíveis de empacotar seis cópias do mesmo livro em quatro caixas idênticas, em que cada caixa pode conter até seis livros?

Solução: Vamos enumerar todas as maneiras de empacotar os livros. Para cada maneira de embalar os livros, listaremos o número de livros na caixa com mais livros, seguido pelo número de livros em cada caixa que contém pelo menos um livro, em ordem decrescente. As maneiras de embalar os livros são

6,
5, 1
4, 2
4, 1, 1
3, 3
3, 2, 1
3, 1, 1, 1
2, 2, 2
2, 2, 1, 1.

Por exemplo, 4, 1, 1 indica que uma caixa contém quatro livros, a segunda caixa contém um livro e a terceira, um livro (e a quarta caixa está vazia). Como enumeramos todas as maneiras de empacotar os seis livros em quatro caixas, vemos que há nove maneiras de embalá-los. ◀

Observe que a distribuição de n objetos idênticos em k caixas idênticas é o mesmo que escrever n como a soma de no máximo k números inteiros positivos em ordem não crescente. Se $a_1 + a_2 + \cdots + a_j = n$, em que $a_1, a_2, \ldots, a_j$ são números inteiros positivos com $a_1 \geq a_2 \geq \cdots \geq a_j$, dizemos que $a_1, a_2, \ldots, a_j$ é uma **partição** do número inteiro positivo n em j números inteiros positivos. Vemos que se $p_k(n)$ é o número de partições de n em no máximo k números inteiros positivos, então há $p_k(n)$ maneiras de distribuir n objetos idênticos em k caixas idênticas. Não existe uma fórmula fechada simples para este número.

Exercícios

1. De quantas maneiras cinco elementos podem ser selecionados em ordem a partir de um conjunto com três elementos, com repetição?
2. De quantas maneiras cinco elementos podem ser selecionados em ordem a partir de um conjunto com cinco elementos, com repetição?
3. Quantas seqüências de seis letras existem?
4. Todo dia um estudante escolhe aleatoriamente um sanduíche de uma pilha de sanduíches. Se há seis tipos de sanduíches, há quantas maneiras possíveis de o estudante escolher os sanduíches para os sete dias da semana, se a ordem em que os sanduíches são escolhidos é relevante?
5. Há quantas maneiras possíveis de designar três empregos para cinco empregados, se cada empregado pode receber mais de um emprego?
6. Há quantas maneiras possíveis de selecionar cinco elementos não ordenados a partir de um conjunto com três elementos, com repetição?
7. Há quantas maneiras possíveis de selecionar três elementos não ordenados a partir de um conjunto com cinco elementos, com repetição?
8. Há quantas maneiras possíveis de escolher uma dúzia de rosquinhas em uma cafeteria com 21 variedades de rosquinhas?
9. Uma loja de pães tem pães de cebola, pães de semente de papoula, pães de ovo, pães de verduras, pães de centeio, pães de gergelim, pães de uva passa e pães simples. Há quantas maneiras possíveis de escolher
   a) seis pães?
   b) uma dúzia de pães?
   c) duas dúzias de pães?
   d) uma dúzia de pães com pelo menos um de cada tipo?
   e) uma dúzia de pães com pelo menos três de ovo e não mais de dois de verdura?
10. Uma padaria tem biscoito simples, biscoito de cereja, biscoito de chocolate, biscoito de amêndoas, biscoito de maçã e biscoito de brócolis. Há quantas maneiras possíveis de escolher

a) uma dúzia de biscoitos?
b) três dúzias de biscoitos?
c) duas dúzias de biscoitos com pelo menos dois de cada tipo?
d) duas dúzias de biscoitos com não mais de dois biscoitos de brócolis?
e) duas dúzias de biscoitos com pelo menos cinco biscoitos de chocolate e pelo menos três biscoitos de amêndoas?
f) duas dúzias de biscoitos com pelo menos um biscoito simples, pelo menos dois biscoitos de cereja, pelo menos três biscoitos de chocolate, pelo menos um biscoito de amêndoa, pelo menos dois biscoitos de maçã e não mais de três biscoitos de brócolis?

11. Há quantas maneiras possíveis de escolher oito moedas de um cofrinho que contém 100 moedas de 10 centavos e 80 moedas de 5 centavos?

12. Há quantas combinações diferentes de moedas de 1, 5, 10, 25 e 50 centavos em um cofrinho se ele contém 20 moedas?

13. Uma editora tem 3000 cópias de um livro de matemática discreta. Há quantas maneiras possíveis de estocar livros se há três armazéns e as cópias do livro são idênticas?

14. Há quantas soluções possíveis para a equação

$$x_1 + x_2 + x_3 + x_4 = 17,$$

em que x_1, x_2, x_3 e x_4 são números inteiros não negativos?

15. Há quantas soluções possíveis para a equação

$$x_1 + x_2 + x_3 + x_4 + x_5 = 21,$$

se x_i, $i = 1, 2, 3, 4, 5$, é um número inteiro não negativo, tal que
a) $x_1 \geq 1$?
b) $x_i \geq 2$ para $i = 1, 2, 3, 4, 5$?
c) $0 \leq x_1 \leq 10$?
d) $0 \leq x_1 \leq 3$, $1 \leq x_2 < 4$ e $x_3 \geq 15$?

16. Há quantas soluções possíveis para a equação

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 29,$$

em que x_i, $i = 1, 2, 3, 4, 5, 6$, é um número inteiro não negativo, tal que
a) $x_i > 1$ para $i = 1, 2, 3, 4, 5, 6$?
b) $x_1 \geq 1$, $x_2 \geq 2$, $x_3 \geq 3$, $x_4 \geq 4$, $x_5 > 5$ e $x_6 \geq 6$?
c) $x_1 \leq 5$?
d) $x_1 < 8$ e $x_2 > 8$?

17. Há quantas seqüências possíveis de 10 dígitos ternários (0, 1 ou 2) que contêm dois 0s, três 1s e cinco 2s?

18. Há quantas seqüências possíveis de 20 dígitos decimais que contêm dois 0s, quatro 1s, três 2s, um 3, dois 4s, três 5s, dois 7s e três 9s?

19. Suponha que uma família grande tenha 14 filhos, incluindo uma dupla de trigêmeos, três grupos de gêmeos e duas crianças. Há quantas maneiras de colocar as crianças em uma fila de cadeiras se os trigêmeos e os gêmeos não podem ser distinguidos um do outro?

20. Há quantas soluções possíveis para a inequação

$$x_1 + x_2 + x_3 \leq 11,$$

em que x_1, x_2 e x_3 são inteiros não negativos? [*Dica:* Introduza uma variável auxiliar x_4, tal que $x_1 + x_2 + x_3 + x_4 = 11$.]

21. Há quantas maneiras possíveis de distribuir seis bolas idênticas em nove caixas distintas?

22. Há quantas maneiras de distribuir 12 bolas idênticas em seis caixas distintas?

23. Há quantas maneiras de distribuir 12 objetos distintos em seis caixas distintas, tal que dois objetos sejam colocados em cada caixa?

24. Há quantas maneiras de distribuir 15 objetos distintos em cinco caixas distintas, para que as caixas tenham um, dois, três, quatro e cinco objetos, respectivamente?

25. Quantos números inteiros positivos menores que 1 000 000 têm a soma de seus dígitos igual a 19?

26. Quantos números inteiros positivos menores que 1 000 000 têm um dígito igual a 9 e têm a soma de dígitos igual a 13?

27. Há 10 questões na prova final de matemática discreta. Há quantas maneiras possíveis de distribuir as notas aos problemas se a soma das notas é 100 e cada questão vale pelo menos 5 pontos?

28. Mostre que há $C(n + r - q_1 - q_2 - \cdots - q_r - 1, n - q_1 - q_2 - \cdots - q_r)$ diferentes seleções não ordenadas de n objetos de r tipos diferentes que incluem pelo menos q_1 objetos do tipo um, q_2 objetos do tipo dois, . . . e q_r objetos do tipo r.

29. Quantas cadeias de bits diferentes podem ser transmitidas se a cadeia deve começar com um bit 1, deve incluir três bits 1 adicionais (para que tenha quatro bits 1 enviados), deve incluir um total de 12 bits 0 e deve ter pelo menos dois bits 0 depois de cada bit 1?

30. Quantas seqüências diferentes podem ser obtidas com as letras da palavra *MISSISSIPPI*, usando-se todas as letras?

31. Quantas seqüências diferentes podem ser obtidas com as letras da palavra *ABRACADABRA*, usando-se todas as letras?

32. Quantas seqüências diferentes podem ser obtidas com as letras de *AARDVARK*, usando-se todas as letras, se todos os três *A*s devem ser consecutivos?

33. Quantas seqüências diferentes podem ser obtidas com as letras de *ORONO*, usando-se até o total das cinco letras?

34. Quantas seqüências com cinco ou mais caracteres podem ser formadas com as letras de *SEERESS*?

35. Quantas seqüências com sete ou mais caracteres podem ser formadas com as letras de *EVERGREEN*?

36. Quantas cadeias de bits diferentes podem ser formadas usando-se seis 1s e oito 0s?

37. Um estudante tem três mangas, dois mamões e dois kiwis. Se ele comer um pedaço de fruta por dia e apenas o tipo da fruta é relevante, de quantas maneiras diferentes essas frutas podem ser consumidas?

38. Uma professora embala sua coleção de 40 artigos de um periódico matemático em quatro caixas com 10 artigos

por caixa. De quantas maneiras ela pode distribuir os periódicos, se

a) cada caixa é numerada, para que elas sejam distintas?

b) as caixas são idênticas, de modo que elas não podem ser distinguidas?

39. Há quantas maneiras possíveis de viajar em um espaço *xyz* com origem (0, 0, 0) ao ponto (4, 3, 5) por meio de passos de uma unidade na direção positiva x, de uma unidade na direção positiva y ou de uma unidade na direção positiva z? (Mover-se nas direções negativas x, y ou z é proibido, portanto os movimentos para trás não são permitidos.)

40. Há quantas maneiras possíveis de andar no espaço *xyzw* a partir da origem (0, 0, 0, 0) ao ponto (4, 3, 5, 4) por meio de passos de uma unidade na direção positiva x, positiva y, positiva z ou positiva w?

41. Há quantas maneiras possíveis de dar sete cartas a cada um dos cinco jogadores de um baralho com 52 cartas?

42. No bridge, as 52 cartas de um baralho são distribuídas a quatro jogadores. Há quantas maneiras diferentes de distribuir as mãos de bridge aos quatro jogadores?

43. Há quantas maneiras de distribuir as mãos de cinco cartas para seis jogadores com um baralho de 48 cartas?

44. De quantas maneiras uma dúzia de livros pode ser colocada em quatro prateleiras distintas

a) se os livros são cópias idênticas do mesmo título?

b) se nenhum dos livros são iguais e as posições dos livros nas prateleiras são relevantes? [*Dica:* Trabalhe como se fossem 12 tarefas, colocando cada livro separadamente. Comece com a seqüência 1, 2, 3, 4 para representar as prateleiras. Represente os livros por b_i, $i = 1, 2, \ldots, 12$. Coloque b_1 à direita de um dos termos 1, 2, 3, 4, que representam as prateleiras. Então coloque sucessivamente b_2, b_3, ... e b_{12}.]

45. De quantas maneiras podem ser colocados n livros em k prateleiras distintas

a) se os livros são cópias idênticas do mesmo título?

b) se nenhum dos livros são iguais e as posições dos livros nas prateleiras são relevantes?

46. Uma prateleira tem 12 livros em uma seqüência. Há quantas maneiras de escolher cinco livros para que dois livros adjacentes não sejam escolhidos? [*Dica:* Represente os livros que são escolhidos por barras e os livros não escolhidos por asteriscos. Conte o número de seqüências de cinco barras e sete asteriscos, sendo que as barras não podem ser adjacentes.]

***47.** Use a regra do produto para demonstrar o Teorema 4, primeiro colocando os objetos na primeira caixa, em seguida colocando-os na segunda caixa e assim por diante.

***48.** Demonstre o Teorema 4 fazendo uma correspondência um para um entre permutações de n objetos com n_i objetos idênticos do tipo i, $i = 1, 2, 3, \ldots, k$, e as distribuições de n objetos em k caixas, tal que n_i objetos sejam colocados na caixa i, $i = 1, 2, 3, \ldots, k$, e então aplique o Teorema 3.

***49.** Neste exercício demonstraremos o Teorema 2 fazendo uma correspondência um para um entre o conjunto de r-combinações com repetição de $S = \{1, 2, 3, \ldots, n\}$ e o conjunto de r-combinações do conjunto $T = \{1, 2, 3, \ldots, n + r - 1\}$.

a) Arranje os elementos em r-combinações, com repetição, de S em uma seqüência crescente $x_1 \leq x_2 \leq \cdots \leq x_r$. Mostre que a seqüência formada pela adição de $k - 1$ ao k-ésimo termo é estritamente crescente. Conclua que essa seqüência é feita de r elementos distintos de T.

b) Mostre que o procedimento descrito em (a) define uma correspondência um para um entre o conjunto de r-combinações, com repetição, de S e o de r-combinações de T. [*Dica:* Mostre que a correspondência pode ser reversa associando a r-combinação $\{x_1, x_2, \ldots, x_r\}$ de T, com $1 \leq x_1 < x_2 < \cdots < x_r \leq n + r - 1$, à r-combinação, com repetição, de S, formada pela subtração $k - 1$ do k-ésimo elemento.]

c) Conclua que há $C(n + r - 1, r)$ r-combinações, com repetição, de um conjunto com n elementos.

50. Há quantas maneiras possíveis de distribuir cinco objetos distintos em três caixas idênticas?

51. Há quantas maneiras possíveis de distribuir seis objetos distintos em quatro caixas idênticas, para que cada uma das caixas tenha pelo menos um objeto?

52. Há quantas maneiras possíveis de colocar cinco empregados temporários em quatro salas idênticas?

53. Há quantas maneiras possíveis de colocar seis empregados temporários em quatro salas idênticas, para que haja pelo menos um empregado em cada uma das salas?

54. Há quantas maneiras possíveis de distribuir cinco objetos idênticos em três caixas idênticas?

55. Há quantas maneiras possíveis de distribuir seis objetos idênticos em quatro caixas idênticas, para que cada uma das caixas tenha pelo menos um objeto?

56. Há quantas maneiras possíveis de embalar oito DVDs idênticos em cinco caixas idênticas, para que cada caixa tenha pelo menos um DVD?

57. Há quantas maneiras possíveis de embalar nove DVDs idênticos em três caixas idênticas, para que cada caixa tenha pelo menos dois DVDs?

58. Há quantas maneiras possíveis de distribuir cinco bolas em sete caixas, se cada caixa deve ter no máximo uma bola e se

a) ambas, bolas e caixas, são distinguíveis?

b) as bolas são distinguíveis, mas as caixas indistinguíveis?

c) as bolas são indistinguíveis, mas as caixas distinguíveis?

d) ambas, bolas e caixas, são indistinguíveis?

59. Há quantas maneiras possíveis de distribuir cinco bolas em três caixas se cada caixa deve conter pelo menos uma bola e se

a) ambas, bolas e caixas, são distintas?

b) as bolas são distintas, mas as caixas idênticas?

c) as bolas são idênticas, mas as caixas distintas?

d) ambas, bolas e caixas, são idênticas?

60. Suponha que uma liga de basquete tenha 32 times, divididos em duas delegações com 16 times cada. Cada delegação é dividida em três divisões. Suponha que a Divisão Centro-Norte tenha cinco times. Cada um dos times da Divisão Centro-Norte joga quatro partidas contra cada um dos outros nesta divisão, três partidas contra cada um dos 11 times restantes da delegação e duas partidas contra cada um dos 16 times da outra delegação. Há quantas ordens de jogos diferentes possíveis para um dos times da Divisão Centro-Norte?

***61.** Suponha que um inspetor de armas deva inspecionar cada um dos cinco lugares diferentes duas vezes, visitando um lugar por dia. O inspetor tem permissão para escolher a ordem em que irá visitar esses lugares, mas não pode visitar X, o lugar mais suspeito, em dois dias consecutivos. Há quantas maneiras possíveis de o inspetor visitar tais lugares?

62. Há quantos termos diferentes possíveis no desenvolvimento de $(x_1 + x_2 + \cdots + x_m)^n$ depois que todos os termos com conjuntos idênticos de expoentes são adicionados?

***63.** Demonstre o **Teorema Multinomial:** Se n é um número inteiro positivo, então

$$(x_1 + x_2 + \cdots + x_m)^n = \sum_{n_1+n_2+\cdots+n_m=n} C(n;n_1,n_2,\cdots,n_m)x_1^{n_1}x_2^{n_2}\cdots x_m^{n_m},$$

em que

$$C(n;n_1,n_2,\cdots,n_m) = \frac{n!}{n_1!n_2!\cdots n_m!}$$

é um **coeficiente multinomial.**

64. Encontre o desenvolvimento de $(x + y + z)^4$.

65. Encontre o coeficiente de $x^3y^2z^5$ em $(x + y + z)^{10}$.

66. Quantos termos existem no desenvolvimento de $(x + y + z)^{100}$?

5.6 Gerando Permutações e Combinações

Introdução

Os métodos para contar vários tipos de permutações e combinações foram descritos nas seções anteriores deste capítulo, mas algumas permutações ou combinações necessitam ser desenvolvidas, não apenas contadas. Considere os três problemas a seguir. Primeiro, suponha que um vendedor deva visitar seis cidades diferentes. Em qual ordem essas cidades devem ser visitadas com o menor tempo de viagem possível? Uma maneira de determinar a melhor ordem é determinar o tempo de viagem para cada uma das $6! = 720$ ordens diferentes nas quais as cidades podem ser visitadas e escolher o menor tempo de viagem. Segundo, suponha que seja dado um conjunto de seis números inteiros positivos e que desejamos encontrar um subconjunto que tenha a soma de seus elementos igual a 100, se tal subconjunto existir. Uma maneira de encontrar esses números é gerar todos os $2^6 = 64$ subconjuntos e checar a soma de seus elementos. Terceiro, suponha que um laboratório tenha 95 empregados. Um grupo de 12 desses empregados, com um conjunto determinado de 25 habilidades, é requisitado para um projeto. (Cada empregado pode ter uma ou mais dessas habilidades.) Uma forma de encontrar tal conjunto de empregados é gerar todos os conjuntos de 12 desses empregados e checar se eles têm as habilidades desejadas. Esses exemplos mostram que geralmente é necessário gerar as permutações e combinações para resolver os problemas.

Gerando Permutações

Links

Qualquer conjunto com n elementos pode ser colocado em correspondência um para um com o conjunto $\{1, 2, 3, \ldots, n\}$. Podemos listar as permutações de qualquer conjunto de n elementos ao gerar permutações dos n menores números inteiros positivos e então colocar novamente esses inteiros com os elementos correspondentes. Muitos algoritmos diferentes têm sido desenvolvidos para a geração das $n!$ permutações deste conjunto. Descreveremos uma delas que é baseada na **ordem lexicográfica** (ou **alfabética)** do conjunto de permutações de $\{1, 2, 3, \ldots, n\}$. Nesta ordem, a permutação $a_1a_2 \cdots a_n$ precede a permutação de $b_1b_2 \cdots b_n$, para qualquer k, com $1 \leq k \leq n$, $a_1 = b_1$, $a_2 = b_2, \ldots, a_{k-1} = b_{k-1}$ e $a_k < b_k$. Em outras palavras, uma permutação do conjunto dos n menores números inteiros positivos precede (na ordem lexicográfica) uma segunda permutação se o número nesta permutação na primeira posição, em que as duas permutações discordam, é menor que o número naquela posição da segunda permutação.

EXEMPLO 1 A permutação 23415 do conjunto $\{1, 2, 3, 4, 5\}$ precede a permutação 23514, pois essas permutações coincidem nas primeiras duas posições, mas o número na terceira posição da primeira permutação, 4, é menor que o número na terceira posição da segunda permutação, 5. De maneira análoga, a permutação 41532 precede a 52143. ◀

Um algoritmo para gerar a permutação de $\{1, 2, \ldots, n\}$ pode ser baseado em um procedimento que constrói a próxima permutação na ordem lexicográfica seguindo uma dada permutação $a_1a_2 \cdots a_n$. Mostraremos como isso pode ser feito. Primeiro, suponha que $a_{n-1} < a_n$. Trocando a_{n-1} por a_n obtemos uma permutação maior. Nenhuma outra permutação é maior que a permutação original e menor que a permutação obtida pela troca de a_{n-1} por a_n, ou seja, não há permutações entre elas, na ordem lexicográfica. Por exemplo, a próxima permutação maior que 234156 é 234165. Por outro lado, se $a_{n-1} > a_n$, então uma permutação maior não pode ser obtida pela troca desses dois últimos termos. Então devemos verificar os três últimos números inteiros na permutação. Se $a_{n-2} < a_{n-1}$, então os últimos três números inteiros da permutação podem ser rearranjados para obter a próxima permutação maior que a original. Coloque o menor dos dois números inteiros, a_{n-1} e a_n, que é maior que a_{n-2}, na posição $n - 2$. Então, coloque o número restante e a_{n-2} nas duas últimas posições em ordem crescente. Por exemplo, a maior permutação depois de 234165 é 234516.

Por outro lado, se $a_{n-2} > a_{n-1}$ (e $a_{n-1} > a_n$), então a maior permutação não pode ser obtida pela troca dos três últimos termos na permutação. Com base nessas observações, um método geral pode ser descrito para produzir a próxima maior permutação em ordem crescente, seguindo uma permutação dada $a_1a_2 \cdots a_n$. Primeiro, encontre os números inteiros a_j e a_{j+1}, com $a_j < a_{j+1}$ e

$$a_{j+1} > a_{j+2} > \cdots > a_n,$$

ou seja, o último par de números inteiros adjacentes da permutação em que o primeiro inteiro no par é menor que o segundo. Então, a próxima permutação maior em ordem lexicográfica é obtida colocando-se na j-ésima posição o menor inteiro entre $a_{j+1}, a_{j+2}, \ldots$ e a_n, que é maior que a_j, e listando-se em ordem crescente o restante dos inteiros $a_j, a_{j+1}, \ldots, a_n$, nas posições de $j + 1$ a n. É fácil ver que não há outra permutação maior que $a_1a_2 \cdots a_n$, mas menor que a nova permutação produzida. (A verificação deste fato foi deixada como exercício para o leitor.)

EXEMPLO 2 Qual é a próxima permutação em ordem lexicográfica depois de 362541?

Exemplos Extras

Solução: O último par de números inteiros a_j e a_{j+1}, em que $a_j < a_{j+1}$, é $a_3 = 2$ e $a_4 = 5$. O menor inteiro à direita de 2, que é maior que 2 na permutação, é $a_5 = 4$. Assim, 4 é colocado na terceira posição. Então, os inteiros 2, 5 e 1 são colocados em ordem crescente nas três últimas posições, resultando 125 nas últimas três posições da permutação. Assim, a próxima permutação é 364125. ◀

Para produzir as $n!$ permutações dos números inteiros $1, 2, 3, \ldots, n$, comece com a menor permutação em ordem lexicográfica, ou seja, $123 \cdots n$, e aplique sucessivamente o processo descrito para produzir a próxima maior permutação $n! - 1$ vezes. Isto resulta em todas as permutações dos n menores números inteiros em ordem lexicográfica.

EXEMPLO 3 Construa as permutações dos números inteiros 1, 2, 3 em ordem lexicográfica.

Solução: Comece com 123. A próxima permutação é obtida pela troca entre 3 e 2 para obter 132. Depois, como $3 > 2$ e $1 < 3$, permute os três números inteiros em 132. Coloque o menor entre 3 e 2 na primeira posição e então coloque 1 e 3 em ordem crescente nas posições 2 e 3 para obter 213. Este é seguido por 231, obtido pela troca entre 1 e 3, porque $1 < 3$. A próxima maior permu-

tação tem 3 na primeira posição, seguido de 1 e 2 em ordem crescente, ou seja, 312. Por fim, troque 1 com 2 para obter a última permutação, 321. ◀

O Algoritmo 1 mostra o procedimento para encontrar a próxima permutação em ordem lexicográfica, depois de uma permutação que não é $n\ n-1\ n-2 \cdots 2\ 1$, que é a maior permutação.

ALGORITMO 1 Geração da Próxima Permutação em Ordem Lexicográfica.

procedure *proxima permutacao*($a_1 a_2 \cdots a_n$: permutação de $\{1, 2, \ldots, n\}$ que não é igual a $n\ n-1 \cdots 2\ 1$)
$j := n - 1$
while $a_j > a_j + 1$
  $j := j - 1$
{j é o maior índice tal que $a_j < a_{j+1}$}
$k := n$
while $a_j > a_k$
  $k := k - 1$
{a_k é o menor número inteiro maior que a_j à direita de a_j }
troquemos a_j e a_k
$r := n$
$s := j + 1$
while $r > s$
begin
  troque a_r e a_s
  $r := r - 1$
  $s := s + 1$
end
{isto coloca a parte final da permutação depois da j-ésima posição em ordem crescente}

Gerando Combinações

Links

Como podemos gerar todas as combinações de elementos de um conjunto finito? Como uma combinação é apenas um subconjunto, podemos usar a correspondência entre subconjuntos de $\{a_1, a_2, \ldots, a_n\}$ e cadeias de bits de extensão n.

Lembre-se de que uma cadeia de bits corresponde a um subconjunto que tem um 1 na posição k se a_k for elemento do subconjunto, e tem um 0 nesta posição se a_k não for um elemento do subconjunto. Se todas as cadeias de bits de extensão n podem ser listadas, então, pela correspondência entre subconjuntos e cadeias de bits, uma lista de todos os subconjuntos é obtida.

Lembre-se de que uma cadeia de bits de extensão n é também uma expansão binária de um número inteiro entre 0 e $2^n - 1$. As 2^n cadeias de bits podem ser listadas em ordem crescente de tamanho assim como os inteiros em suas expansões binárias. Para produzir todas as expansões binárias de extensão n, comece com a seqüência 000 . . . 00, com n zeros. Então encontre sucessivamente a próxima expansão até que a cadeia 111 . . . 11 seja obtida. Em cada etapa, a próxima expansão binária é encontrada localizando-se a primeira posição à direita que não seja 1, e, em seguida, mudando todos os 1s à direita desta posição para 0s e trocando o primeiro 0 (da direita) por 1.

EXEMPLO 4 Encontre a próxima cadeia de bits depois de 10 0010 0111.

Solução: O primeiro bit da direita que não for um 1 é o quarto bit da direita. Mudamos esse bit para 1 e todos os bits subseqüentes para 0. Isto produzirá a próxima cadeia de bits, 10 0010 1000. ◀

O processo de produção da próxima maior cadeia de bits depois de $b_{n-1}b_{n-2}\ldots b_1b_0$ é dado pelo Algoritmo 2.

ALGORITMO 2 Geração da Maior Cadeia de Bits Subseqüente.

procedure *proxima cadeia de bit* ($b_{n-1}\, b_{n-2}\ldots b_1b_0$: cadeia diferente de 11...11)
$i := 0$
while $b_i = 1$
begin
  $b_i := 0$
  $i := i + 1$
end
$b_i := 1$

A seguir daremos um algoritmo para a geração de r-combinações do conjunto $\{1, 2, 3, \ldots, n\}$. Uma r-combinação pode ser representada por uma seqüência que contém os elementos no subconjunto em ordem crescente. As r-combinações podem ser listadas usando-se a ordem lexicográfica dessas seqüências. A próxima combinação depois de $a_1a_2\cdots a_r$ pode ser obtida da seguinte forma: primeiro, localize o último elemento a_i na seqüência, tal que $a_i \neq n - r + i$. Depois, substitua a_i por $a_i + 1$ e a_j por $a_i + j - i + 1$, para $j = i + 1, i + 2, \ldots, r$. Resta ao leitor mostrar que isto produz a maior combinação subseqüente em ordem lexicográfica. Este procedimento é ilustrado com o Exemplo 5.

EXEMPLO 5 Encontre a maior 4-combinação subseqüente do conjunto $\{1, 2, 3, 4, 5, 6\}$ depois de $\{1, 2, 5, 6\}$.

Solução: O último termo entre os termos a_i, com $a_1 = 1$, $a_2 = 2$, $a_3 = 5$ e $a_4 = 6$, tal que $a_i \neq 6 - 4 + i$, é $a_2 = 2$. Para obter a maior 4-combinação subseqüente, adicione 1 a a_2 para obter $a_2 = 3$. Então, o conjunto $a_3 = 3 + 1 = 4$ e $a_4 = 3 + 2 = 5$. Assim, a maior 4-combinação subseqüente é $\{1, 3, 4, 5\}$. ◀

O Algoritmo 3 mostra os pseudocódigos para esse procedimento.

ALGORITMO 3 Geração da Próxima *r*-Combinação em Ordem Lexicográfica.

procedure *proxima r-combinacao*($\{a_1, a_2, \ldots, a_r\}$: o subconjunto adequado de
    $\{1, 2, \ldots, n\}$ diferente de $\{n - r + 1, \ldots, n\}$ com
    $a_1 < a_2 < \cdots < a_r$)
$i := r$
while $a_i = n - r + i$
  $i := i - 1$
$a_i := a_i + 1$
for $j := i + 1$ **to** r
  $a_j := a_i + j - i$

Exercícios

1. Coloque as permutações de {1, 2, 3, 4, 5} em ordem lexicográfica: 43521, 15432, 45321, 23451, 23514, 14532, 21345, 45213, 31452, 31542.
2. Coloque as permutações de{1,2,3,4,5,6} em ordem lexicográfica: 234561, 231456, 165432, 156423, 543216, 541236, 231465, 314562, 432561, 654321, 654312, 435612.
3. Encontre a maior permutação subseqüente em ordem lexicográfica das permutações abaixo.
   a) 1432 **b)** 54123
   c) 12453 **d)** 45231
   e) 6714235 **f)** 31528764
4. Encontre a maior permutação subseqüente em ordem lexicográfica das permutações abaixo.
   a) 1342 **b)** 45321
   c) 13245 **d)** 612345
   e) 1623547 **f)** 23587416
5. Use o Algoritmo 1 para gerar as 24 permutações dos quatro primeiros números inteiros positivos em ordem lexicográfica.
6. Use o Algoritmo 2 para listar todos os subconjuntos do conjunto {1, 2, 3, 4}.
7. Use o Algoritmo 3 para listar todas as 3-combinações de {1, 2, 3, 4, 5}.
8. Mostre que o Algoritmo 1 produz a maior permutação subseqüente em ordem lexicográfica.
9. Mostre que o Algoritmo 3 produz a maior r-combinação subseqüente em ordem lexicográfica, dada uma r-combinação.
10. Desenvolva um algoritmo para a geração de r-permutações de um conjunto com n elementos.
11. Liste todas as 3-permutações de {1, 2, 3, 4, 5}.

Os exercícios restantes desta seção desenvolvem outro algoritmo para a geração de permutações de $\{1, 2, 3, \ldots, n\}$. Esse algoritmo é baseado nas expansões de números inteiros de Cantor. Todo inteiro não negativo menor que $n!$ tem uma única expansão de Cantor

$$a_1 1! + a_2 2! + \cdots + a_{n-1}(n-1)!,$$

em que a_i é um número inteiro não negativo não excedente a i, para $i = 1, 2, \ldots, n-1$. Os números inteiros $a_1, a_2, \ldots, a_{n-1}$ são chamados de **dígitos de Cantor** desse inteiro.

Dada uma permutação de $\{1, 2, \ldots, n\}$, considere a_{k-1}, $k = 2, 3, \ldots, n$, como o número de inteiros menores que k que são seguidos por k na permutação. Por exemplo, na permutação 43215, a_1 é o número de inteiros menores que 2 que o acompanha, então $a_1 = 1$. Do mesmo modo, para este exemplo, $a_2 = 2$, $a_3 = 3$ e $a_4 = 0$. Considere a função do conjunto de permutações $\{1, 2, 3, \ldots, n\}$ para o conjunto de inteiros não negativos menores que $n!$ que levam uma permutação a um inteiro que tenha $a_1, a_2, \ldots, a_{n-1}$, definido desta maneira, como dígitos de Cantor.

12. Encontre os números inteiros que correspondem às permutações abaixo.
    a) 246531 **b)** 12345 **c)** 654321
*13. Mostre que a correspondência descrita aqui é uma bijeção entre o conjunto de permutações de $\{1, 2, 3, \ldots, n\}$ e os números inteiros não negativos menores que $n!$.
14. Encontre a permutação de {1, 2, 3, 4, 5} que corresponde aos números inteiros abaixo, respeitando a correspondência entre as expansões de Cantor e as permutações, como descrito no preâmbulo do Exercício 12.
    a) 3 **b)** 89 **c)** 111
15. Desenvolva um algoritmo para produzir todas as permutações de um conjunto de n elementos, com base na correspondência descrita no preâmbulo do Exercício 12.

Termos-chave e Resultados

TERMOS

combinatória: o estudo dos arranjos de objetos

enumeração: a contagem dos arranjos de objetos

diagrama de árvore: um diagrama composto por uma raiz, galhos que saem dessa raiz e outros galhos que saem do final de alguns galhos

permutação: um arranjo ordenado de elementos de um conjunto

***r*-permutação:** um arranjo ordenado de r elementos de um conjunto

***P*(*n*, *r*):** o número de r-permutações de um conjunto com n elementos

***r*-combinação:** uma seleção não ordenada de r elementos de um conjunto

***C*(*n*, *r*):** o número de r-combinações de um conjunto com n elementos

$\binom{n}{r}$ **(coeficiente binomial):** também o número de r-combinações de um conjunto com n elementos

demonstração combinatorial (ou combinatória) de uma identidade: uma demonstração que usa argumentos de contagem para demonstrar que os dois lados de uma identidade fazem a contagem dos mesmos objetos, mas de formas diferentes

triângulo de Pascal: uma representação de coeficientes binomiais em que a i-ésima linha do triângulo contém $\binom{i}{j}$ para $j = 0, 1, 2, \ldots, i$

RESULTADOS

a regra do produto: uma técnica básica de contagem, que estabelece que o número de maneiras de fazer um procedimento que consiste em duas sub-tarefas é o produto do número de maneiras de fazer a primeira tarefa e o número de maneiras de fazer a segunda tarefa depois que a primeira foi executada.

a regra da soma: uma técnica básica de contagem, que estabelece que o número de maneiras de realizar uma tarefa de uma ou duas

formas é a soma do número de maneiras de realizar essas tarefas se elas não puderem ser feitas simultaneamente

o princípio da casa dos pombos: Quando mais de k objetos são colocados em k caixas, deverá haver uma caixa que tenha mais de um objeto.

a generalização do princípio da casa dos pombos: Quando N objetos são colocados em k caixas, deverá haver um caixa que tenha pelo menos $\lceil N/k \rceil$ objetos.

$$P(n,r) = \frac{n!}{(n-r)!}$$

$$C(n,r) = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

identidade de Pascal: $\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$

o binômio de Newton: $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$

Há n^r r-permutações de um conjunto com n elementos com repetição.

Há $C(n+r-1, r)$ r-combinações de um conjunto com n elementos com repetição.

Há $n!/(n_1!\, n_2! \cdots n_k!)$ permutações de n objetos, em que há n_i objetos idênticos do tipo i, para $i = 1, 2, 3, \ldots, k$.

O algoritmo para gerar as permutações do conjunto $\{1, 2, \ldots, n\}$.

Questões de Revisão

1. Explique como as regras da soma e do produto podem ser usadas para encontrar o número de cadeias de bits com comprimento não excedente a 10.

2. Explique como encontrar o número de cadeias de bits de extensão não excedente a 10 que tenham pelo menos um bit 0.

3. a) Como a regra do produto pode ser usada para encontrar o número de funções de um conjunto com m elementos para um conjunto com n elementos?

b) Quantas funções existem de um conjunto com cinco elementos para um conjunto com 10 elementos?

c) Como a regra do produto pode ser usada para encontrar o número de funções injetoras de um conjunto com m elementos para um conjunto com n elementos?

d) Quantas funções injetoras existem de um conjunto com cinco elementos para um conjunto com 10 elementos?

e) Quantas funções sobrejetivas existem de um conjunto com cinco elementos para um conjunto com 10 elementos?

4. Como é possível encontrar o número de resultados de uma eliminatória entre dois times, em que o primeiro que vencer quatro jogos vence a eliminatória?

5. Como é possível encontrar o número da cadeia de bits de extensão 10 que ou começa com 101 ou termina com 010?

6. a) Descreva o princípio da casa dos pombos.

b) Explique como o princípio da casa dos pombos pode ser usado para mostrar que entre 11 números inteiros quaisquer, pelo menos dois devem ter o último dígito igual.

7. a) Estabeleça a generalização do princípio da casa dos pombos.

b) Explique como a generalização do princípio da casa dos pombos pode ser usada para mostrar que entre 91 números inteiros quaisquer, pelo menos 10 têm o mesmo dígito final.

8. a) Qual a diferença entre uma r-combinação e uma r-permutação de um conjunto com n elementos?

b) Derive uma equação que relaciona o número de r-combinações e o número de r-permutações de um conjunto com n elementos.

c) Há quantas maneiras possíveis de escolher seis estudantes a partir de uma classe com 25 para montar um comitê?

d) Há quantas maneiras possíveis de escolher seis estudantes a partir de uma classe com 25 para ocuparem seis cargos executivos diferentes em um comitê?

9. a) O que é o triângulo de Pascal?

b) Como uma linha do triângulo de Pascal pode ser produzida a partir de outra acima dela?

10. Qual o significado de uma demonstração combinatorial de uma identidade? Qual a diferença entre essa demonstração e a algébrica?

11. Explique como demonstrar a identidade de Pascal usando um argumento combinatorial.

12. a) Descreva o binômio de Newton.

b) Explique como demonstrar o binômio de Newton usando um argumento combinatorial.

c) Encontre o coeficiente de $x^{100}y^{101}$ na expansão de $(2x + 5y)^{201}$.

13. a) Explique como encontrar a fórmula para o número de maneiras de escolher r objetos a partir de n objetos com repetição, sendo a ordem não relevante.

b) Há quantas maneiras possíveis de escolher uma dúzia de objetos a partir de cinco tipos diferentes de objetos, se os objetos do mesmo tipo são idênticos?

c) Há quantas maneiras possíveis de escolher uma dúzia de objetos a partir de cinco tipos diferentes de objetos, se deve haver pelo menos três objetos do primeiro tipo?

d) Há quantas maneiras possíveis de escolher uma dúzia de objetos a partir de cinco tipos diferentes de objetos se não pode haver mais de quatro objetos do primeiro tipo?

e) Há quantas maneiras possíveis de escolher uma dúzia de objetos a partir de cinco tipos diferentes de objetos se deve haver pelo menos dois objetos do primeiro tipo mas não mais que três do segundo tipo?

14. a) Considere n e r como números inteiros positivos. Explique por que o número de soluções da equação $x_1 + x_2$

$+ \cdots + x_n = r$, em que x_i é um número inteiro não negativo para $i = 1, 2, 3, \ldots, n$, é igual ao número de r-combinações de um conjunto com n elementos.

b) Quantas soluções com números inteiros não negativos são possíveis para a equação $x_1 + x_2 + x_3 + x_4 = 17$?

c) Quantas soluções com números inteiros positivos são possíveis para a equação do item (b)?

15. a) Derive uma fórmula para o número de permutações de n objetos de k tipos diferentes, em que há n_1 objetos idênticos do tipo um, n_2 objetos idênticos do tipo dois, ... e n_k objetos idênticos do tipo k.

b) Há quantas maneiras possíveis de ordenar as letras da palavra *INDISCREETNESS*?

16. Descreva um algoritmo para gerar todas as permutações do conjunto dos n menores números inteiros positivos.

17. a) Há quantas maneiras possíveis de distribuir cinco cartas para seis jogadores a partir de um baralho com 52 cartas?

b) Há quantas maneiras possíveis de distribuir n objetos distintos em k caixas distintas, para que n_i objetos sejam colocados na caixa i?

18. Descreva um algoritmo para gerar todas as combinações do conjunto com os n menores números inteiros positivos.

Exercícios Complementares

1. Há quantas maneiras possíveis de escolher 6 itens de 10 itens distintos, quando

a) os itens escolhidos são ordenados e não há repetição?

b) os itens escolhidos são ordenados e há repetição?

c) os itens escolhidos não são ordenados e não há repetição?

d) os itens escolhidos não são ordenados e há repetição?

2. Há quantas maneiras possíveis de escolher 10 itens de 6 itens distintos, quando

a) os itens escolhidos são ordenados e não há repetição?

b) os itens escolhidos são ordenados e há repetição?

c) os itens escolhidos não são ordenados e não há repetição?

d) os itens escolhidos não são ordenados e há repetição?

3. Um teste contém 100 questões do tipo falso/verdadeiro. De quantas maneiras um estudante pode responder às questões desse teste se nenhuma questão pode ficar em branco?

4. Quantas cadeias de bits de extensão 10 ou começam com 000 ou terminam com 1111?

5. Quantas cadeias de bits de extensão 10 sobre o alfabeto $\{a, b, c\}$ têm exatamente três as ou quatro bs?

6. Os números de telefone internos em um sistema de um campus possuem cinco dígitos, com o primeiro dígito diferente de zero. Quantos números de telefone diferentes o sistema suporta?

7. Uma sorveteria tem 28 sabores diferentes, 8 coberturas diferentes e 12 acompanhamentos.

a) De quantas maneiras uma taça com três bolas de sorvete pode ser montada, em que cada sabor pode ser colocado mais de uma vez e a ordem das bolas não é relevante?

b) Quantos tipos diferentes de sundaes pequenos podem ser montados se um sundae pequeno tem uma bola de sorvete, uma cobertura e um acompanhamento?

c) Quantos tipos diferentes de sundaes grandes podem ser montados se um sundae grande tem três bolas de sorvete, em que cada sabor pode ser colocado mais de uma vez e a ordem das bolas não é relevante; dois tipos de cobertura, em que cada cobertura pode ser usada apenas uma vez e a ordem não é relevante; e três acompanhamentos, em que cada acompanhamento pode ser usado apenas uma vez e a ordem não é relevante?

8. Quantos números inteiros positivos menores que 1000

a) têm três dígitos decimais?

b) têm um número ímpar de dígitos decimais?

c) têm pelo menos um dígito decimal igual a 9?

d) não têm dígitos decimais ímpares?

e) têm dois dígitos decimais consecutivos iguais a 5?

f) são palíndromos (ou seja, podem ser lidos de frente para trás e de trás para frente)?

9. Quando os números de 1 a 1000 são escritos em notação decimal, quantos dos dígitos abaixo são usados?

a) 0 **b)** 1

c) 2 **d)** 9

10. Há 12 signos do zodíaco. Quantas pessoas são necessárias para garantir que pelo menos seis delas tenham o mesmo signo?

11. Uma empresa de biscoitos da sorte faz 213 biscoitos diferentes. Um estudante almoça em um restaurante que compra biscoitos dessa empresa. Qual o maior número de vezes possível que o estudante pode comer no restaurante sem pegar quatro vezes o mesmo tipo de biscoito?

12. Quantas pessoas são necessárias para garantir que pelo menos duas tenham nascido no mesmo dia da semana e no mesmo mês (talvez em anos diferentes)?

13. Mostre que, dado qualquer conjunto de 10 números inteiros positivos não excedentes a 50, existem pelo menos dois subconjuntos diferentes com cinco elementos que têm a mesma soma.

14. Uma embalagem com figurinhas de beisebol contém 20 figurinhas. Quantas embalagens devem ser compradas para

garantir que duas cartas nessas embalagens sejam idênticas, se há no total 550 figurinhas diferentes?

15. **a)** Quantas cartas devem ser escolhidas de um baralho para garantir que pelo menos dois ases sejam escolhidos?
b) Quantas cartas devem ser escolhidas de um baralho para garantir que pelo menos dois ases e dois naipes sejam escolhidos?
c) Quantas cartas devem ser escolhidas de um baralho para garantir que pelo menos duas cartas do mesmo naipe sejam escolhidas?
d) Quantas cartas devem ser escolhidas de um baralho para garantir que pelo menos duas cartas de dois naipes diferentes sejam escolhidas?

***16.** Mostre que em qualquer conjunto de $n + 1$ números inteiros positivos não excedentes a $2n$ deve haver dois números primos entre si.

***17.** Mostre que em uma seqüência com m números inteiros existe um ou mais termos consecutivos com a soma divisível por m.

18. Mostre que se cinco pontos forem escolhidos em uma quadra com comprimento de lado 2, então pelo menos dois desses pontos estão distantes não mais que $\sqrt{2}$.

19. Mostre que uma expansão decimal de um número racional deve ter uma seqüência repetida infinitamente a partir de um ponto.

20. Uma vez que um vírus infecta um computador pessoal via e-mail, ele manda uma cópia dele mesmo a 100 endereços que se encontram na caixa de entrada desse computador. Qual o número máximo de computadores diferentes que este computador pode infectar no tempo que leva para a mensagem infectada ser reenviada cinco vezes?

21. Há quantas maneiras possíveis de escolher doze rosquinhas a partir de 20 variedades
a) se não houver duas rosquinhas da mesma variedade?
b) se todas as rosquinhas forem da mesma variedade?
c) se não houver restrições?
d) se houver pelo menos duas variedades?
e) se deverá haver pelo menos seis rosquinhas de amora?
f) se não puder haver mais que seis rosquinhas de amora?

22. Encontre n se
a) $P(n, 2) = 110$.
b) $P(n, n) = 5040$.
c) $P(n, 4) = 12P(n, 2)$.

23. Encontre n se
a) $C(n, 2) = 45$.
b) $C(n, 3) = P(n, 2)$.
c) $C(n, 5) = C(n, 2)$.

24. Mostre que se n e r são números inteiros não negativos e $n \geq r$, então
$$P(n + 1, r) = P(n, r)(n + 1)/(n + 1 - r).$$

***25.** Suponha que S seja um conjunto com n elementos. Quantos pares ordenados (A, B) são possíveis, considerando-se que A e B são subconjuntos de S com $A \subseteq B$? [*Dica:* Mostre que cada elemento de S pertence a A, $B - A$ ou $S - B$.]

26. Dê uma demonstração combinatorial do Corolário 2 da Seção 5.4, a partir da correspondência entre os subconjuntos de um conjunto com um número par de elementos e os subconjuntos desse conjunto com um número ímpar de elementos. [*Dica:* Pegue um elemento a no conjunto. Faça a correspondência, colocando a no subconjunto, se ainda não estiver, ou tirando-o se ele estiver no subconjunto.]

27. Considere n e r como números inteiros não negativos com $r < n$. Mostre que
$$C(n, r - 1) = C(n + 2, r + 1) - 2C(n + 1, r + 1) + C(n, r + 1).$$

28. Demonstre, usando a indução matemática, que $\Sigma_{j=2}^{n} C(j,2) = C(n + 1, 3)$ sempre que n for um número inteiro maior que 1.

29. Mostre que se n for um número inteiro, então
$$\sum_{k=0}^{n} 3^k \binom{n}{k} = 4^n.$$

30. Neste exercício vamos derivar uma fórmula para a soma dos quadrados dos n menores números inteiros positivos. Contaremos o número de triplas (i, j, k) considerando-se que i, j e k são números inteiros, sendo que $0 \leq i < k$, $0 \leq j < k$ e $1 \leq k \leq n$ de duas maneiras.
a) Mostre que há k^2 de tais triplas com base k. Conclua que há $\sum_{k=1}^{n} k^2$ triplas.
b) Mostre que o número de triplas com $0 \leq i < j < k$ e o número de triplas com $0 \leq j < i < k$ é igual a $C(n + 1, 3)$.
c) Mostre que o número de triplas com $0 \leq i = j < k$ é igual a $C(n + 1, 2)$.
d) Combine os itens (a), (b) e (c) e conclua que
$$\sum_{k=1}^{n} k^2 = 2C(n + 1, 3) + C(n + 1, 2) = n(n + 1)(2n + 1)/6.$$

***31.** Quantas seqüências de bits de extensão n, em que $n \geq 4$, contêm duas ocorrências de 01?

32. Considere S como um conjunto. Dizemos que uma coleção de subconjuntos $A_1, A_2, \ldots, A_n$, cada um com d elementos, em que $d \geq 2$, é 2-*colorável*, se for possível designar a cada elemento de S uma de duas cores diferentes, tal que cada subconjunto A_i tenha elementos que sejam coloridos com cada cor. Considere $m(d)$ como o maior número inteiro, tal que toda a coleção menor que $m(d)$ conjuntos com d elementos seja 2-colorável.
a) Mostre que a coleção de todos os subconjuntos com d elementos de um conjunto S com $2d - 1$ elementos não é 2-colorável.
b) Mostre que $m(2) = 3$.
****c)** Mostre que $m(3) = 7$. [*Dica:* Mostre que a coleção {1, 3, 5}, {1, 2, 6}, {1, 4, 7}, {2, 3, 4}, {2, 5, 7},{3, 6, 7}, {4, 5, 6} não é 2-colorável. Então, mostre que todas as coleções de seis conjuntos com três elementos cada são 2-coloráveis.]

33. Um professor escreve 20 questões de múltipla escolha, cada uma com as possíveis respostas a, b, c ou d, para um teste de matemática discreta. Se o número de questões com as alternativas a, b, c e d tiverem como resposta 8, 3, 4 e 5, respectivamente, quantas respostas corretas diferentes são possíveis, se as questões podem ser colocadas em qualquer ordem?

34. Há quantos arranjos diferentes possíveis de oito pessoas sentarem-se em uma mesa redonda, em que dois arranjos são considerados os mesmos se um for obtido pelo outro a partir de uma rotação?

35. Há quantas maneiras possíveis de distribuir 24 estudantes como cinco conselheiros da faculdade?

36. Há quantas maneiras possíveis de escolher uma dúzia de maçãs de uma caixa que contém 20 maçãs idênticas Delicious, 20 maçãs idênticas Macintosh e 20 maçãs idênticas Granny Smith, se pelo menos três de cada tipo devem ser escolhidas?

37. Quantas soluções são possíveis para a equação $x_1 + x_2 + x_3 = 17$, em que x_1, x_2 e x_3 são números inteiros não negativos com

a) $x_1 > 1, x_2 > 2$ e $x_3 > 3$?

b) $x_1 < 6$ e $x_3 > 5$?

c) $x_1 < 4, x_2 < 3$ e $x_3 > 5$?

38. a) Quantas seqüências diferentes podem ser feitas com a palavra *PEPPERCORN* quando todas as letras forem usadas?

b) Quantas dessas seqüências começam e terminam com a letra *P*?

c) Em quantas dessas seqüências as três letras *P* aparecem em posição consecutiva?

39. Quantos subconjuntos de um conjunto com 10 elementos

a) têm menos que cinco elementos?

b) têm mais que sete elementos?

c) têm um número ímpar de elementos?

40. Uma testemunha de um acidente em que o motorista fugiu conta à polícia que a placa do carro, que contém três letras seguidas de três números, começa com as letras *AS* e contém os números 1 e 2. Quantas placas diferentes cabem nesta descrição?

41. Há quantas maneiras possíveis de colocar n objetos idênticos em m contêineres distintos para que nenhum contêiner fique vazio?

42. Há quantas maneiras possíveis de sentar seis garotos e oito garotas em uma fila de cadeiras para que nenhum garoto sente ao lado de outro?

43. Há quantas maneiras possíveis de distribuir seis objetos em cinco caixas se

a) os objetos e as caixas são distintos?

b) os objetos são distintos, mas as caixas idênticas?

c) os objetos são idênticos, mas as caixas distintas?

d) os objetos e as caixas são idênticos?

44. Há quantas maneiras possíveis de distribuir cinco objetos em seis caixas se

a) os objetos e as caixas são distintos?

b) os objetos são distintos, mas as caixas idênticas?

c) os objetos são idênticos, mas as caixas distintas?

d) os objetos e as caixas são idênticos?

45. Desenvolva um algoritmo para gerar todas as r-permutações de um conjunto finito com repetição.

46. Desenvolva um algoritmo para gerar todas as r-combinações de um conjunto finito com repetição.

***47.** Mostre que se m e n são números inteiros com $m \geq 3$ e $n \geq 3$, então $R(m, n) \leq R(m, n-1) + R(m-1, n)$.

***48.** Mostre que $R(3, 4) \geq 7$ mostrando que em um grupo de seis pessoas, em que duas pessoas são amigos ou inimigos, não há necessariamente três amigos mútuos ou quatro inimigos mútuos.

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dado um número inteiro positivo n e um número inteiro não negativo que não exceda a n, encontre o número de r-permutações e r-combinações de um conjunto com n elementos.

2. Dado os números inteiros positivos n e r, encontre o número de r-permutações com repetição e r-combinações com repetição de um conjunto com n elementos.

3. Dada uma seqüência de números inteiros positivos, encontre a maior subseqüência crescente e a maior decrescente desta seqüência.

***4.** Dada uma equação $x_1 + x_2 + \cdots + x_n = C$, em que C é uma constante e $x_1, x_2, \ldots, x_n$ são números inteiros não negativos, liste todas as soluções.

5. Dado um número inteiro positivo n, liste todas as permutações do conjunto $\{1, 2, 3, \ldots, n\}$ em ordem lexicográfica.

6. Dado um número inteiro positivo n e um inteiro não negativo r não excedente a n, liste todas as r-combinações do conjunto $\{1, 2, 3, \ldots, n\}$ em ordem lexicográfica.

7. Dado um número inteiro positivo n e um inteiro não negativo r não excedente a n, liste todas as r-permutações do conjunto $\{1, 2, 3, \ldots, n\}$ em ordem lexicográfica.

8. Dado um número inteiro positivo n, liste todas as combinações do conjunto $\{1, 2, 3, \ldots, n\}$.

9. Dados os números inteiros positivos n e r, liste todas as r-permutações, com repetição, do conjunto $\{1, 2, 3, \ldots, n\}$.

10. Dados os números inteiros positivos n e r, liste todas as r-combinações, com repetição, do conjunto $\{1, 2, 3, \ldots, n\}$.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Encontre o número de possíveis resultados de uma eliminatória com dois times, quando o vencedor é o primeiro time que vencer 5 de 9, 6 de 11, 7 de 13 e 8 de 15 partidas.
2. Quais coeficientes binomiais são ímpares? Você pode formular uma conjectura baseada em uma evidência numérica?
3. Verifique que $C(2n, n)$ é divisível pelo quadrado de um primo, quando $n \neq 1, 2$ ou 4, para tantos números inteiros positivos n quantos você puder. [O teorema que diz que $C(2n, n)$ é divisível pelo quadrado de um primo com $n \neq 1, 2$ ou 4 foi demonstrado em 1996 por Andrew Granville e Olivier Ramaré. Essa demonstração foi baseada em uma conjectura feita em 1980 por Paul Erdös e Ron Graham.]
4. Encontre quantos números n você puder, que sejam inteiros ímpares menores que 200 e para os quais $C(n, \lfloor n/2 \rfloor)$ não é divisível pelo quadrado de um número primo. Formule uma conjectura baseada em sua evidência.
5. * Para cada número inteiro menor que 100, determine se $C(2n, n)$ é divisível por 3. Você pode formular uma conjectura que nos diga que para número inteiro n, o coeficiente binomial $C(2n, n)$ é divisível por 3, baseado nos dígitos da expansão de n em base três?
6. Gere todas as permutações de um conjunto com oito elementos.
7. Gere todas as 6-permutações de um conjunto com nove elementos.
8. Gere todas as combinações de um conjunto com oito elementos.
9. Gere todas as 5-combinações, com repetição, de um conjunto com sete elementos.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Descreva os primeiros usos do princípio da casa dos pombos feitos por Dirichlet e outros matemáticos.
2. Discuta qual plano telefônico atual pode ser expandido para acomodar a rápida demanda por mais números de telefone. (Veja se você pode encontrar algumas das propostas vindas da indústria de telecomunicações.) Para cada novo plano telefônico, mostre como encontrar o número de diferentes números de telefone que ele suporta.
3. Muitas identidades combinatórias são descritas neste livro. Encontre algumas fontes de tais identidades e descreva identidades combinatórias importantes além das já introduzidas neste livro. Dê algumas demonstrações representativas, incluindo as combinatórias, de algumas dessas identidades.
4. Descreva os diferentes modelos utilizados na distribuição de partículas em mecânica estatística, incluindo os modelos de Maxwell–Boltzmann, Bose–Einstein e Fermi–Dirac. Em cada caso, descreva as técnicas de contagem usadas no modelo.
5. Defina os números de Stirling de primeira ordem e descreva algumas de suas propriedades e as identidades que eles satisfazem.
6. Defina os números de Stirling de segunda ordem e descreva algumas de suas propriedades e as identidades que eles satisfazem.
7. Descreva as últimas descobertas de valores e aproximações dos números de Ramsey.
8. Descreva maneiras adicionais de gerar todas as permutações de um conjunto com n elementos entre aqueles encontrados na Seção 5.6. Compare esses algoritmos e os algoritmos descritos no texto e nos exercícios da Seção 5.6 em termos de sua complexidade computacional.
9. Descreva pelo menos uma maneira de gerar todas as partições de um número inteiro positivo n. (Veja o Exercício 47 da Seção 4.3.)

CAPÍTULO

6 Probabilidade Discreta

6.1 Uma Introdução à Probabilidade Discreta
6.2 Teoria da Probabilidade
6.3 Teorema de Bayes
6.4 Valor Esperado e Variância

Combinatória e teoria da probabilidade têm origens comuns. A teoria da probabilidade foi a primeira a ser desenvolvida há mais de 300 anos, quando alguns jogos de azar foram analisados. Embora a teoria da probabilidade tenha sido originariamente inventada para estudar as apostas, desenvolve agora um papel essencial em uma grande variedade de disciplinas. Por exemplo, a teoria da probabilidade é extensamente aplicada ao estudo da genética, no qual pode ser usada para entender a herança de traços genéticos. Claro que a probabilidade persiste como uma parte muito popular da matemática por sua aplicabilidade aos jogos, que continua sendo uma possibilidade de ganho fácil.

Na ciência da computação, a teoria da probabilidade desenvolve um importante papel no estudo da complexidade de algoritmos. Em particular, as idéias e técnicas da teoria da probabilidade são usadas para determinar a complexidade do caso médio de algoritmos. Algoritmos probabilísticos podem ser usados para resolver muitos problemas que não podem ser facilmente ou totalmente resolvidos por algoritmos determinísticos. Em um algoritmo probabilístico, em vez de seguir os mesmos passos quando é dada a mesma entrada, como um algoritmo determinístico faz, o algoritmo faz uma ou mais escolhas aleatórias, que podem acarretar resultados diferentes. Em combinatória, a teoria da probabilidade pode até ser usada para mostrar que os objetos com determinadas propriedades existem. O método de probabilidade, uma técnica em combinatória introduzida por Paul Erdös e Alfréd Rényi, mostra que um objeto com determinada propriedade existe, indicando que há uma probabilidade alta de um objeto construído aleatoriamente ter essa propriedade. A teoria da probabilidade pode nos ajudar a responder questões que envolvem incerteza, tais como determinar se devemos rejeitar uma mensagem de e-mail recebida como spam com base nas palavras que aparecem na mensagem.

6.1 Uma Introdução à Probabilidade Discreta

Introdução

A teoria da probabilidade data de antes do século XVII, quando o matemático francês Blaise Pascal determinou como estudar a vitória em jogos populares baseado em resultados do lançamento de um par de dados repetidamente. No século XVIII, o matemático francês Laplace, que também estudou os jogos, definiu a probabilidade de um evento como o número de resultados favoráveis dividido pelo número de resultados possíveis. Por exemplo, a probabilidade de sair um número ímpar quando um dado é jogado, é o número de resultados favoráveis, ou seja, o número de possibilidades de ser ímpar, dividido pelo número de resultados possíveis, ou seja, o número de resultados diferentes ao lançar um dado. Há um total de seis possibilidades, ou seja, 1, 2, 3, 4, 5 e 6, e exatamente três dessas possibilidades podem ocorrer, ou seja, 1, 3 e 5. Assim, a probabilidade de sair um número ímpar quando um dado é jogado é $3/6 = 1/2$. (Note que assumimos que todas as possibilidades são iguais, ou, em outras palavras, que o dado não é viciado.)

Nesta seção, nos restringiremos aos experimentos que têm possibilidades finitas e iguais de acontecer. Isso nos permite usar a definição de Laplace da probabilidade de um evento. Continuaremos nosso estudo de probabilidade na Seção 6.2, onde estudaremos experimentos com possibilidades finitas, mas que não são necessariamente iguais. Na Seção 6.2, introduziremos também alguns conceitos-chave na teoria da probabilidade, incluindo a probabilidade condicional, independência de eventos e variáveis aleatórias. Na Seção 6.4, introduziremos os conceitos de valor esperado e variância de uma variável aleatória.

Probabilidade Finita

Um **experimento** é um procedimento que produz um resultado de um conjunto de resultados possíveis. O **espaço amostral** do experimento é o conjunto de possíveis resultados. Um **evento** é um subconjunto do espaço amostral. A definição de probabilidade de Laplace para um evento com resultados possíveis finitos será estabelecida agora.

DEFINIÇÃO 1 Se S é um espaço amostral finito de resultados igualmente prováveis e E é um evento, ou seja, um subconjunto de S, então a *probabilidade* de E é $p(E) = \frac{|E|}{|S|}$.

Os exemplos 1 a 8 ilustram como a probabilidade de um evento é encontrada.

EXEMPLO 1 Uma urna contém quatro bolas azuis e cinco bolas vermelhas. Qual é a probabilidade de uma bola escolhida na urna ser azul?

Exemplos Extras

Solução: Para calcular a probabilidade, note que há nove escolhas possíveis e quatro dessas escolhas podem ser uma bola azul. Assim, a probabilidade de uma bola azul ser escolhida é $4/9$. ◄

EXEMPLO 2 Qual é a probabilidade, quando dois dados são jogados, de a soma dos números dos dados ser 7?

Solução: Há, no total, 36 possibilidades iguais quando dois dados são jogados. (A regra do produto pode ser usada para mostrar isso; como cada dado tem seis possibilidades, o número total de possibilidades quando dois dados são jogados é $6^2 = 36$.) Há seis possibilidades favoráveis, ou seja, (1, 6), (2, 5), (3, 4), (4, 3), (5, 2) e (6, 1), em que os valores do primeiro e do segundo dado são representados por um par ordenado. Assim, a probabilidade de um sete sair quando dois dados não-viciados forem jogados é $6/36 = 1/6$. ◄

Links

As loterias são extremamente populares. Podemos facilmente computar a probabilidade de vencer em diferentes tipos das loterias.

EXEMPLO 3 Em uma loteria, os jogadores ganham o maior prêmio quando acertam quatro dígitos, na ordem correta, quatro dígitos selecionados aleatoriamente por um processo mecânico. Ganha-se o me-

Links

PIERRE-SIMON LAPLACE (1749–1827) Pierre-Simon Laplace era de origem humilde, da Normandia. Em sua infância, ele foi educado em uma escola beneditina. Aos 16 anos, entrou para a Universidade de Caen para estudar teologia. Entretanto, ele logo percebeu que seu real interesse era a matemática. Depois de completar seus estudos, ele foi nomeado professor provisório em Caen e, em 1769, tornou-se professor de matemática na Escola Militar de Paris.

Laplace é mais conhecido por suas contribuições em mecânica celestial, o estudo dos movimentos dos corpos celestes. Seu *Traité du Mécanique Céleste* é considerado um dos maiores trabalhos científicos do início do século XIX. Laplace foi um dos fundadores da teoria da probabilidade e fez muitas contribuições para a matemática estatística. Seu trabalho nesta área está documentado em seu livro *Thèorie Analytique des Probabilitées*, no qual ele definiu a probabilidade de um evento como a razão entre os números de possibilidades favoráveis e o número total de possibilidades de um experimento.

Laplace era famoso por sua flexibilidade política. Ele foi fiel, nesta ordem, à República Francesa, a Napoleão e ao Rei Luis XVIII. Esta flexibilidade permitiu-lhe ser produtivo antes, durante e depois da Revolução Francesa.

nor prêmio se apenas três números forem acertados. Qual é a probabilidade de um jogador ganhar o prêmio máximo? Qual a probabilidade de um jogador ganhar o menor prêmio?

Solução: Há apenas uma maneira de escolher todos os quatro números corretamente. Pela regra do produto, há $10^4 = 10\,000$ maneiras de escolher os quatro números. Assim, a probabilidade que um jogador tem de ganhar o maior prêmio é $1/10\,000 = 0{,}0001$.

Os jogadores ganham o menor prêmio quando eles escolhem corretamente três dos quatro números. Exatamente um número deve ser errado para termos três números corretos, mas não todos os quatro corretos. Pela regra da soma, para encontrar o número de maneiras de escolher exatamente três números corretamente, adicionamos o número de maneiras de escolher quatro dígitos que combinam com os números escolhidos, menos na i-ésima posição, para $i = 1, 2, 3, 4$.

Para contar o número de sucessos com o primeiro número incorreto, note que há nove escolhas possíveis para o primeiro dígito (todos, menos o dígito correto) e uma escolha para cada um dos outros dígitos, ou seja, os dígitos certos para tais posições. Assim, há nove maneiras de escolher os quatro dígitos em que o primeiro é incorreto, mas os três últimos são corretos. Da mesma forma, há nove maneiras de escolhê-los em que o segundo dígito é incorreto, nove com o terceiro dígito incorreto e nove com o quarto número incorreto. Assim, há um total de 36 maneiras de escolher quatro números com três dos quatro corretos. Então, a probabilidade de um jogador ganhar o menor prêmio é $36/10\,000 = 9/2\,500 = 0{,}0036$. ◀

EXEMPLO 4 Existem muitas loterias com prêmios enormes para as pessoas que escolherem corretamente um conjunto de seis números dos primeiros n números inteiros positivos, em que n está normalmente entre 30 e 60. Qual a probabilidade que uma pessoa tem de escolher seis números corretos dentre 40 números?

Solução: Há apenas uma combinação vencedora. O número total de maneiras de escolher seis números dentre 40 é

$$C(40, 6) = \frac{40!}{34!6!} = 3\,838\,380.$$

Conseqüentemente, a probabilidade de escolher uma combinação vencedora é $1/3\,838\,380 \approx 0{,}00000026$. (Aqui, o símbolo $\approx$ significa "aproximadamente igual a".) ◀

Links

O pôquer e outros jogos de cartas estão crescendo em popularidade. Para vencer nesses jogos, ajuda saber a probabilidade de mãos diferentes. Podemos encontrar a probabilidade de uma determinada mão que aparece em jogos de cartas usando as técnicas desenvolvidas até aqui. Um baralho contém 52 cartas. Há 13 tipos diferentes de cartas, com quatro cartas de cada tipo. (Entre os termos comumente usados, além de "tipo", temos "rank", "valor da carta", "denominação" e "valor".) Essas cartas são dois, três, quatro, cinco, seis, sete, oito, nove, dez, valete, rainha, rei e ás. Há também quatro naipes: espadas, ouros, copas e paus, cada um com 13 cartas, com uma carta de cada tipo em um naipe. Em muitos jogos de pôquer, uma mão consiste em cinco cartas.

EXEMPLO 5 Encontre a probabilidade de uma mão com cinco cartas em um jogo de pôquer ter quatro cartas do mesmo tipo.

Solução: Pela regra do produto, o número de mãos com cinco cartas que contêm quatro cartas de um tipo é o produto do número de maneiras de escolher um tipo de carta, o número de maneiras de escolher quatro desse tipo e o número de maneiras de escolher a quinta carta. Isto é,

$$C(13, 1)C(4, 4)C(48, 1).$$

Pelo Exemplo 11 da Seção 5.3, há $C(52, 5)$ mãos diferentes de cinco cartas. Assim, a probabilidade de uma mão conter quatro cartas de um tipo é

$$\frac{C(13,1)C(4,4)C(48,1)}{C(52,5)} = \frac{13 \cdot 1 \cdot 48}{2\,598\,960} \approx 0,00024. \quad \blacktriangleleft$$

EXEMPLO 6 Qual é a probabilidade de uma mão de pôquer conter um *full house*, ou seja, três cartas de um tipo e dois de outro?

Solução: Pela regra do produto, o número de mãos com um *full house* é o produto do número de maneiras de escolher dois tipos em ordem, o número de maneiras de escolher três cartas de quatro para o primeiro tipo e o número de maneiras de escolher duas das quatro do segundo tipo. (Note que a ordem desses tipos é relevante, porque, por exemplo, três rainhas e dois ases são diferentes de três ases e duas rainhas.) Vemos que o número de mãos com um *full house* é

$$P(13, 2)C(4, 3)C(4, 2) = 13 \cdot 12 \cdot 4 \cdot 6 = 3744.$$

Como há 2 598 960 mãos de pôquer, a probabilidade de um *full house* é

$$\frac{3744}{2\,598\,960} \approx 0,0014. \quad \blacktriangleleft$$

EXEMPLO 7 Qual é a probabilidade de os números 11, 4, 17, 39 e 23 serem retirados em ordem de uma caixa com 50 bolas marcadas com os números 1, 2, . . . , 50 se (a) a bola escolhida não retorna à caixa antes de as próximas bolas serem retiradas e (b) a bola escolhida é recolocada na caixa antes de as próximas bolas serem retiradas?

Solução: (a) Pela regra do produto, há $50 \cdot 49 \cdot 48 \cdot 47 \cdot 46 = 254.251.200$ maneiras de selecionar as bolas, já que cada vez que uma bola é retirada, há uma bola a menos para ser escolhida. Conseqüentemente, a probabilidade de 11, 4, 17, 39 e 23 serem tirados na ordem é de $1/254.251.200$. Este é um exemplo de **amostra sem reposição.**

(b) Pela regra do produto, há $50^5 = 312.500.000$ maneiras de selecionar as bolas já que há 50 bolas para serem escolhidas toda vez que uma bola for retirada. Conseqüentemente, a probabilidade de 11, 4, 17, 39 e 23 serem retirados nesta ordem é de $1/312.500.000$. Este é um exemplo de **amostra com reposição.** $\blacktriangleleft$

A Probabilidade de Combinações de Eventos

Podemos usar técnicas de contagem para encontrar a probabilidade de eventos derivados de outros eventos.

TEOREMA 1 Considere E como um evento em um espaço amostral S. A probabilidade do evento $\overline{E}$, o evento complementar de E, é dada por

$$p(\overline{E}) = 1 - p(E).$$

Demonstração: Para encontrar a probabilidade do evento $\overline{E}$, note que $|\overline{E}| = |S| - |E|$. Assim,

$$p(\overline{E}) = \frac{|S| - |E|}{|S|} = 1 - \frac{|E|}{|S|} = 1 - p(E). \quad \triangleleft$$

Há uma estratégia alternativa para encontrar a probabilidade de um evento quando um método direto não funciona muito bem. Em vez de determinar a probabilidade do evento, a probabilidade de seu complemento pode ser encontrada. Isso é visto facilmente no Exemplo 8.

EXEMPLO 8 Uma seqüência de 10 bits é construída aleatoriamente. Qual é a probabilidade de pelo menos um desses bits ser 0?

Solução: Considere E como o evento em que pelo menos um dos 10 bits é 0. Então, $\overline{E}$ é o evento em que todos os bits são 1. Como o espaço amostral S é o conjunto de todas as cadeias de bits de extensão 10, temos que

$$p(E) = 1 - p(\overline{E}) = 1 - \frac{|\overline{E}|}{|S|} = 1 - \frac{1}{2^{10}}$$
$$= 1 - \frac{1}{1024} = \frac{1023}{1024}.$$

Assim, a probabilidade de a cadeia de bits conter pelo menos um bit 0 é $1023/1024$. É um pouco difícil encontrar essa probabilidade direta sem usar o Teorema 1. ◀

Podemos encontrar também a probabilidade da união de dois eventos.

TEOREMA 2 Considere E_1 e E_2 como eventos no espaço amostral S. Então,

$$p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2).$$

Demonstração: Usando a fórmula dada na Seção 2.2 para o número de elementos na união de dois conjuntos, temos que

$$|E_1 \cup E_2| = |E_1| + |E_2| - |E_1 \cap E_2|.$$

Assim,

$$p(E_1 \cup E_2) = \frac{|E_1 \cup E_2|}{|S|}$$
$$= \frac{|E_1| + |E_2| - |E_1 \cap E_2|}{|S|}$$
$$= \frac{|E_1|}{|S|} + \frac{|E_2|}{|S|} - \frac{|E_1 \cap E_2|}{|S|}$$
$$= p(E_1) + p(E_2) - p(E_1 \cap E_2).$$

◁

EXEMPLO 9 Qual é a probabilidade de um número inteiro positivo selecionado aleatoriamente a partir do conjunto dos números inteiros positivos não excedentes a 100 ser divisível por 2 ou 5?

Exemplos Extras *Solução:* Considere E_1 como o evento em que o número inteiro selecionado é divisível por 2, e considere E_2 como o evento em que ele é divisível por 5. Então, $E_1 \cup E_2$ é o evento em que o

inteiro é divisível por 2 ou 5. Assim, $E_1 \cap E_2$ é o evento em que o número escolhido é divisível por 2 e por 5, ou equivalentemente, é divisível por 10. Como $|E_1| = 50$, $|E_2| = 20$ e $|E_1 \cap E_2| = 10$, temos que

$$p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2)$$
$$= \frac{50}{100} + \frac{20}{100} - \frac{10}{100} = \frac{3}{5}.$$

◀

Raciocínio Probabilístico

Um problema comum é determinar qual dos dois eventos é mais provável de acontecer. Analisar as probabilidades de tais eventos pode ser bem difícil. O Exemplo 10 descreve um problema desse tipo. Ele discute um famoso problema originário do programa de auditório americano chamado *Let's Make a Deal.*

EXEMPLO 10 **O Problema das Três Portas de Monty Hall** Suponha que você seja um participante de um programa de auditório. Você tem a chance de ganhar um grande prêmio. Você tem de escolher uma das três portas para abrir; o maior prêmio está atrás de uma dessas portas, e as outras portas não têm nada. Uma vez escolhida a porta, o apresentador do programa, que sabe o que está atrás de cada porta, prossegue. Primeiro, se você escolher ou não a porta vencedora, ele abre uma das outras duas portas que ele sabe que não tem nada (selecionada aleatoriamente se as duas portas não tiverem nada). Então, ele pergunta se você gostaria de trocar de porta. Qual estratégia você deve usar? Você deve mudar de porta ou manter sua escolha original?

Links

Solução: A probabilidade de você escolher a porta certa (antes de o apresentador abrir uma porta e perguntar a você se quer trocar) é $1/3$, porque as três portas são iguais e têm a mesma probabilidade de serem a correta. A probabilidade da porta correta não muda depois que o apresentador abre uma das outras portas, porque ele sempre vai abrir uma porta que não contém o prêmio.

A probabilidade de você escolher incorretamente é a probabilidade de o prêmio estar atrás de uma das duas portas que você não escolheu. Conseqüentemente, a probabilidade de você selecionar incorretamente é $2/3$. Se você escolheu a porta errada, quando o apresentador abrir uma porta e mostrar que o prêmio não está ali, o prêmio estará atrás da outra porta. Você sempre vai ganhar se a sua escolha inicial for incorreta e você trocar as portas. Assim, trocando as portas, a probabilidade de você ganhar é de $2/3$. Em outras palavras, você deve sempre trocar as portas quando tiver a chance de fazê-lo. Isso dobra a probabilidade de você vencer. (Um tratamento mais rigoroso deste enigma pode ser encontrado no Exercício 15 da Seção 6.3.) ◀

Exercícios

1. Qual é a probabilidade de uma carta selecionada em um baralho ser um ás?
2. Qual é a probabilidade de sair seis em um dado quando ele é lançado?
3. Qual é a probabilidade de um número inteiro selecionado aleatoriamente a partir dos primeiros 100 números inteiros ser ímpar?
4. Qual é a probabilidade de um dia selecionado aleatoriamente do ano (a partir dos 366 dias possíveis) ser em abril?
5. Qual é a probabilidade de a soma dos números de dois dados ser par quando eles são lançados?
6. Qual é a probabilidade de uma carta selecionada na mesa ser um ás ou uma carta de copas?
7. Qual é a probabilidade de sair seis caras quando uma moeda for lançada seis vezes?
8. Qual é a probabilidade de uma mão de pôquer com cinco cartas conter o ás de copas?
9. Qual é a probabilidade de uma mão de pôquer com cinco cartas não ter a rainha de copas?
10. Qual é a probabilidade de uma mão de pôquer com cinco cartas ter dois de ouros e três de espadas?
11. Qual é a probabilidade de uma mão de pôquer com cinco cartas conter dois de ouros, três de espadas, seis de copas, dez de paus e rei de copas?
12. Qual é a probabilidade de uma mão de pôquer com cinco cartas ter exatamente um ás?

13. Qual é a probabilidade de uma mão de pôquer com cinco cartas ter pelo menos um ás?

14. Qual é a probabilidade de uma mão de pôquer com cinco cartas ter cartas de cinco tipos diferentes?

15. Qual é a probabilidade de uma mão de pôquer com cinco cartas ter dois pares (ou seja, dois de cada dois tipos e uma quinta carta de um terceiro tipo)?

16. Qual é a probabilidade de uma mão de pôquer com cinco cartas ter um *flush*, ou seja, cinco cartas do mesmo naipe?

17. Qual é a probabilidade de uma mão de pôquer com cinco cartas ter um *straight*, ou seja, cinco cartas de tipos consecutivos? (Note que um ás pode ser considerado ou a menor carta de uma seqüência A-2-3-4-5 ou a maior carta da seqüência 10-J-Q-K-A.)

18. Qual é a probabilidade de uma mão de pôquer com cinco cartas ter uma *straight flush*, ou seja, cinco cartas do mesmo naipe de tipos consecutivos?

***19.** Qual é a probabilidade de uma mão de pôquer com cinco cartas ter cartas de cinco tipos diferentes e não ter nem um *flush* nem um *straight*?

20. Qual é a probabilidade de uma mão de pôquer com cinco cartas ter um *royal flush*, ou seja, 10, valete, rainha, rei e ás do mesmo naipe?

21. Qual é a probabilidade de nunca sair um número par quando um dado é jogado seis vezes?

22. Qual é a probabilidade de um número inteiro positivo não excedente a 100, selecionado aleatoriamente, ser divisível por 3?

23. Qual é a probabilidade de um número inteiro positivo não excedente a 100, selecionado aleatoriamente, ser divisível por 5 ou por 7?

24. Encontre a probabilidade de ganhar na loteria escolhendo os seis números inteiros corretos, em que a ordem em que esses inteiros são escolhidos não é relevante, a partir dos números inteiros positivos não excedentes a

a) 30. **b)** 36. **c)** 42. **d)** 48.

25. Encontre a probabilidade de ganhar na loteria escolhendo os seis números inteiros corretos, em que a ordem em que esses inteiros são escolhidos não é relevante, a partir dos números inteiros positivos não excedentes a

a) 50. **b)** 52. **c)** 56. **d)** 60.

26. Encontre a probabilidade de não selecionar nenhum dos seis números inteiros corretamente, em que a ordem em que esses inteiros são escolhidos não é relevante, a partir dos números inteiros positivos não excedentes a

a) 40. **b)** 48. **c)** 56. **d)** 64.

27. Encontre a probabilidade de selecionar corretamente apenas um dos seis números inteiros, em que a ordem em que esses inteiros são escolhidos não é relevante, a partir dos números inteiros positivos não excedentes a

a) 40. **b)** 48. **c)** 56. **d)** 64.

28. Na superloteria, um jogador escolhe 7 números de uma lista de 80 números inteiros positivos. Qual é a probabilidade de uma pessoa ganhar o prêmio máximo escolhendo 7 números entre os 11 selecionados aleatoriamente por um computador?

29. Em uma superloteria, os jogadores ganham uma fortuna se eles acertarem os oito números escolhidos por um computador a partir de números inteiros positivos não excedentes a 100. Qual é a probabilidade de um jogador ganhar nessa superloteria?

30. Qual é a probabilidade de um jogador ganhar um prêmio oferecido para a escolha correta de cinco (mas não seis) de seis números escolhidos aleatoriamente a partir de números inteiros entre 1 e 40, inclusive?

31. Suponha que 100 pessoas participem de um concurso e que os vencedores sejam selecionados aleatoriamente para os prêmios de primeiro, segundo e terceiro lugar. Qual é a probabilidade de Michele ganhar um desses prêmios se ela é um dos participantes?

32. Suponha que 100 pessoas participem de um concurso e que os ganhadores sejam escolhidos aleatoriamente para os prêmios de primeiro, segundo e terceiro lugar. Qual é a probabilidade de Kumar, Janice e Pedro ganharem um prêmio qualquer se cada um participou do concurso?

33. Qual é a probabilidade de Abby, Barry e Sylvia ganharem o primeiro, o segundo e o terceiro prêmio, respectivamente, se 200 pessoas participarem de um concurso e

a) ninguém puder ganhar mais que um prêmio?

b) for permitido ganhar mais de um prêmio?

34. Qual é a probabilidade de Bo, Colleen, Jeff e Rohini ganharem o primeiro, o segundo, o terceiro e o quarto prêmio, respectivamente, se 50 pessoas participarem de um concurso e

a) ninguém puder ganhar mais que um prêmio?

b) for permitido ganhar mais de um prêmio?

35. Na roleta, uma roda com 38 números é girada. Destes, 18 são vermelhos e 18 são pretos. Os outros dois números, que não são nem preto nem vermelho, são 0 e 00. A probabilidade de a roda, quando girada, parar sobre qualquer número determinado é $1/38$.

a) Qual é a probabilidade de a roda parar sobre um número vermelho?

b) Qual é a probabilidade de a roda parar sobre um número preto duas vezes em seqüência?

c) Qual é a probabilidade de a roda parar sobre 0 ou 00?

d) Qual é a probabilidade de, em cinco rodadas, a roda nunca parar sobre 0 ou 00?

e) Qual é a probabilidade de a roda parar sobre um número entre 1 e 6, inclusive, em um giro, mas não parar entre eles no próximo giro?

36. O que é mais provável: sair um total de 8 quando dois dados são lançados ou sair um total de 8 quando três dados são lançados?

37. O que é mais provável: sair um total de 9 quando dois dados são lançados ou sair um total de 9 quando três dados são lançados?

38. Dois eventos E_1 e E_2 são chamados de **independentes** se $p(E_1 \cap E_2) = p(E_1)p(E_2)$. Para cada um dos pares seguintes de eventos, que são subconjuntos do conjunto de todos resultados possíveis quando uma moeda é lançada três vezes, determine se eles são ou não independentes.

a) E_1: a primeira moeda virar coroa; E_2: a segunda moeda virar cara.

b) E_1: a primeira moeda virar coroa; E_2: duas, e não três, virarem cara.

c) E_1: a segunda moeda virar coroa; E_2: duas, e não três, virarem cara.

(Vamos estudar a independência de eventos com mais profundidade na Seção 6.2.)

39. Explique o que está errado com a proposição de que no Problema de Três Portas de Monty Hall a probabilidade de o prêmio estar atrás da primeira porta que você selecionar e a probabilidade de o prêmio estar atrás de uma das outras duas portas que a Monty não abre é $1/2$, porque há duas portas restantes.

40. Suponha que em vez de três portas, haja quatro portas no Problema Monty Hall. Qual é a probabilidade de você ganhar não trocando uma vez com o apresentador, que sabe o que está por trás de cada porta, abre uma porta sem nada e lhe dá a chance de trocar de portas? Qual é a probabilidade de você vencer trocando a porta que você escolheu por uma das duas portas restantes entre as três que você não escolheu?

41. Este problema foi proposto por Chevalier de Méré e foi resolvido por Blaise Pascal e Pierre de Fermat.

a) Encontre a probabilidade de sair pelo menos um seis quando um dado é lançado quatro vezes.

b) Encontre a probabilidade de saírem dois seis em dois dados, pelo menos uma vez, quando eles são lançados 24 vezes. Responda à pergunta que Chevalier de Méré fez a Pascal sobre essa probabilidade ser maior que $1/2$.

c) É mais provável que um seis saia pelo menos uma vez quando um dado é lançado quatro vezes ou que dois seis saiam em dois dados, pelo menos uma vez, quando eles foram lançados 24 vezes?

6.2 Teoria da Probabilidade

Introdução

Links

Na Seção 6.1, introduzimos a noção da probabilidade de um evento. (Lembre-se que um evento é um subconjunto dos resultados possíveis de um experimento.) Definimos a probabilidade de um evento E como Laplace fez, ou seja,

$$p(E) = \frac{|E|}{|S|},$$

o número de resultados favoráveis em E dividido pelo número total de resultados. Essa definição assume que todos os resultados são igualmente possíveis. Entretanto, muitos experimentos têm resultados que não são igualmente possíveis. Por exemplo, uma moeda pode ser viciada para que vire cara duas vezes mais que coroa. Da mesma forma, a possibilidade que tem a entrada de uma busca linear ser um determinado elemento em uma lista, ou não estar na lista, depende de como sua entrada é construída. Como podemos modelar a possibilidade de eventos em tais situações? Nesta seção mostraremos como definir as possibilidades de resultados para estudar as probabilidades dos experimentos quando os resultados não são igualmente possíveis.

Suponha que uma moeda não-viciada é lançada quatro vezes e, na primeira vez, ela vira cara. Dada essa informação, qual é a probabilidade de virar cara três vezes? Para responder isso e questões parecidas, introduziremos o conceito de *probabilidade condicional.* Saber que o primeiro lançamento é cara muda a probabilidade de virar cara três vezes? Se não, esses dois eventos são chamados de *independentes,* um conceito estudado mais tarde nesta seção.

Muitas questões possuem determinado valor numérico associado ao resultado de um experimento. Por exemplo, quando lançada uma moeda 100 vezes, qual é a probabilidade de aparecerem 40 caras? Quantas caras esperávamos que aparecessem? Nesta seção, introduziremos as *variáveis aleatórias,* que são funções que associam valores numéricos a resultados de experimentos.

Links

NOTA HISTÓRICA Chevalier de Méré era um nobre francês, um famoso apostador e bon-vivant. Ele teve sucesso como apostador em jogos com probabilidades um pouco maiores que $1/2$ (tal como ter pelo menos um seis em quatro lançamentos de um dado). Sua correspondência com Pascal perguntando sobre a probabilidade de ter dois seis, pelo menos uma vez, em um par de dados lançado 24 vezes, permitiu o desenvolvimento da teoria da probabilidade. De acordo com um relato, Pascal escreveu a Fermat falando de Chevalier algo do tipo "Ele é um bom rapaz, mas não é matemático".

Determinando Probabilidades

Considere S como o espaço amostral de um experimento com um número finito, ou contável, de resultados. Atribuímos uma probabilidade $p(s)$ para cada resultado s. Necessitamos que duas condições sejam respeitadas:

(i) $0 \leq p(s) \leq 1$ para cada $s \in S$

e

$$(ii) \sum_{s \in S} p(s) = 1.$$

A condição (i) estabelece a probabilidade de cada resultado ser um número real não negativo e não maior que 1. A condição (ii) estabelece que a soma das probabilidades de todos os resultados possíveis deve ser 1; ou seja, quando realizamos o experimento, é certeza que um desses resultados apareça. (Note que quando um espaço amostral é infinito, $\Sigma_{s \in S}\, p(s)$ é uma série infinita convergente.) Isso é uma generalização da definição de Laplace, na qual para cada um dos n resultados é determinada uma probabilidade de $1/n$. De fato, as condições (i) e (ii) são respeitadas quando a definição de Laplace de probabilidades de resultados com possibilidades iguais é usada e S é finito. (Veja o Exercício 4.)

Note que quando há n resultados possíveis, $x_1, x_2, \ldots, x_n$, as duas condições a serem seguidas são

(i) $0 \leq p(x_i) \leq 1$ para $i = 1, 2, \ldots, n$

e

$$(ii) \sum_{i=1}^{n} p(x_i) = 1.$$

A função p a partir do conjunto de todos os resultados do espaço amostral S é chamada de **distribuição de probabilidade.**

Para modelar um experimento, a probabilidade $p(s)$ atribuída a um resultado s deve ser igual ao limite do número de vezes que s aparece dividido pelo número de vezes que o experimento é realizado, fazendo este número aumentar muito. (Vamos assumir que todos os experimentos discutidos têm resultados previstos, portanto esse limite existe. Também assumimos que os resultados de sucessivos testes de um experimento não dependem de resultados passados.)

Lembre-se: Não discutiremos as probabilidades de eventos quando o conjunto de resultados não for finito, ou contável, tal como quando o resultado for um experimento que pode ser um número real. Nesses casos, o cálculo integral é normalmente necessário para estudar a probabilidade de eventos.

Podemos modelar os experimentos nos quais tais resultados são igualmente possíveis, ou não, escolhendo a função apropriada $p(s)$, como ilustra o Exemplo 1.

EXEMPLO 1 Quais probabilidades devemos determinar para os resultados H (cara) e T (coroa) quando uma moeda não-viciada for lançada? Quais probabilidades devem ser determinadas para esses resultados quando a moeda é viciada para que as caras saiam duas vezes mais que as coroas?

Solução: Para uma moeda não-viciada, a probabilidade de sair cara quando uma moeda é lançada é igual à probabilidade de sair coroa, para que os resultados sejam igualmente possíveis. Conseqüentemente, determinamos a probabilidade $1/2$ para cada um desses dois resultados possíveis, ou seja, $p(H) = p(T) = 1/2$.

Para a moeda viciada, temos

$$p(H) = 2p(T).$$

Como

$$p(H) + p(T) = 1,$$

temos que

$$2p(T) + p(T) = 3p(T) = 1.$$

Concluímos que $p(T) = 1/3$ e $p(H) = 2/3$. ◀

DEFINIÇÃO 1 Suponha que S seja um conjunto com n elementos. A *distribuição uniforme* atribui a probabilidade $1/n$ para cada elemento de S.

Agora definiremos a probabilidade de um evento como a soma das probabilidades dos resultados neste evento.

DEFINIÇÃO 2 A *probabilidade* do evento E é a soma das probabilidades dos resultados em E, ou seja,

$$p(E) = \sum_{s \in E} p(s).$$

(Note que quando E é um conjunto infinito, $\Sigma_{s \in E} p(s)$ é uma série infinita convergente.)

Note que quando há n resultados no evento E, ou seja, se $E = \{a_1, a_2, \ldots, a_n\}$, então $p(E) = \sum_{i=1}^{n} p(a_i)$. Note também que a distribuição uniforme atribui a um evento a mesma probabilidade que a definição original de probabilidade de Laplace atribui a esse evento. O experimento de selecionar um elemento a partir de um espaço amostral com uma distribuição uniforme é chamado de seleção **aleatória** de um elemento de S.

EXEMPLO 2 Suponha que um dado seja viciado (ou falso) para que o 3 apareça duas vezes mais que os outros números, mas que os outro cinco resultados são igualmente possíveis. Qual é a probabilidade de um número ímpar aparecer quando o dado é lançado?

Solução: Queremos encontrar a probabilidade do evento $E = \{1, 3, 5\}$. Pelo Exercício 2 no final desta seção, temos que

$$p(1) = p(2) = p(4) = p(5) = p(6) = 1/7; p(3) = 2/7.$$

Temos que

$$p(E) = p(1) + p(3) + p(5) = 1/7 + 2/7 + 1/7 = 4/7.$$ ◀

Quando os eventos são igualmente possíveis e há um número finito de resultados possíveis, a definição de probabilidade de um evento dada nesta seção (Definição 2) concorda com a definição de Laplace (Definição 1 da Seção 6.1). Para ver isto, suponha que haja n resultados igualmente possíveis; cada resultado possível tem probabilidade $1/n$, porque a soma de suas probabilidades é 1. Suponha que o evento E tenha m resultados. De acordo com a Definição 2,

$$p(E) = \sum_{i=1}^{m} \frac{1}{n} = \frac{m}{n}.$$

Como $|E| = m$ e $|S| = n$, temos que

$$p(E) = \frac{m}{n} = \frac{|E|}{|S|}.$$

Esta é a definição de Laplace para a probabilidade do evento E.

Combinações de Eventos

As fórmulas para as probabilidades de combinações de eventos da Seção 6.1 continuam válidas quando usamos a Definição 2 para definir a probabilidade de um evento. Por exemplo, o Teorema 1 da Seção 6.1 mostra que

$$p(\overline{E}) = 1 - p(E),$$

em que $\overline{E}$ é o evento complementar do evento E. Esta equação também se mantém quando a Definição 2 é utilizada. Para ver isso, note que como a soma das probabilidades dos n resultados possíveis é 1 e cada resultado está em E ou em $\overline{E}$, mas não em ambos, temos

$$\sum_{s \in S} p(s) = 1 = p(E) + p(\overline{E}).$$

Assim, $p(\overline{E}) = 1 - p(E)$.

Considerando a definição de Laplace, pelo Teorema 2 da Seção 6.1, temos

$$p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2)$$

sempre que E_1 e E_2 forem eventos em um espaço amostral S. Isso também é válido quando definimos a probabilidade de um evento como fizemos nesta seção. Para ver isso, note que $p(E_1 \cup E_2)$ é a soma das probabilidades dos resultados em $E_1 \cup E_2$. Quando um resultado x está em um dos E_1 e E_2, mas não em ambos, $p(x)$ aparece em uma das somas para $p(E_1)$ e $p(E_2)$. Quando um resultado x está em ambos, E_1 e E_2, $p(x)$ aparece na soma para $p(E_1)$, na soma para $p(E_2)$ e na soma para $p(E_1 \cap E_2)$, assim, ele aparece $1 + 1 - 1 = 1$ vez do lado direito. Conseqüentemente, o lado esquerdo e o lado direito da equação são iguais.

A probabilidade da união de eventos disjuntos pode ser encontrada usando o Teorema 1.

TEOREMA 1 Se $E_1, E_2, \ldots$ for uma seqüência de eventos disjuntos em um espaço amostral S, então

$$p\left(\bigcup_i Ei\right) = \sum_i p(Ei).$$

(Note que este teorema é aplicado quando a seqüência $E_1, E_2, \ldots$ consiste em um número finito ou infinito contável de eventos disjuntos.)

Deixamos a demonstração do Teorema 1 para o leitor (veja os exercícios 36 e 37).

Probabilidade Condicional

Links

Suponha que joguemos uma moeda três vezes, e que todas as oito possibilidades sejam iguais. Além disso, suponha que saibamos que o evento F, que na primeira jogada vira coroa, aconteça. Dada essa informação, qual é a probabilidade de o evento E, que é um número ímpar de coroas, acontecer? Como a primeira jogada é coroa, há apenas quatro resultados possíveis: *TTT*, *TTH*, *THT* e *THH*, em que *H e T* representam caras e coroas, respectivamente. Um número ímpar de coroas aparece apenas para os resultados *TTT* e *THH*. Como os oito resultados têm probabilidade igual, cada um dos quatro resultados possíveis, dado que F aconteça, deve ter uma probabilidade igual de $1/4$. Isso sugere que devemos determinar a probabilidade de $2/4 = 1/2$ para E, dado que F ocorreu. Essa probabilidade é chamada de **probabilidade condicional** de E dado F.

Em geral, para encontrar a probabilidade condicional de E dado F, usamos F como espaço amostral. Para um resultado de E acontecer, esse resultado deve também pertencer a $E \cap F$. Com esta motivação, desenvolvemos a Definição 3.

DEFINIÇÃO 3 Considere E e F como eventos com $p(F) > 0$. A *probabilidade condicional* de E dado F, indicada por $p(E \mid F)$, é definida como

$$p(E \mid F) = \frac{p(E \cap F)}{p(F)}.$$

EXEMPLO 3 Exemplos Extras

Uma cadeia de bits de extensão quatro é construída aleatoriamente para que todas as 16 cadeias de bits de extensão quatro sejam igualmente possíveis. Qual é a probabilidade de ela ter pelo menos dois zeros consecutivos, dado que seu primeiro bit é 0? (Assumimos que bits 0 e bits 1 são igualmente possíveis.)

Solução: Considere E como o evento de uma cadeia de bits de extensão quatro que tem pelo menos dois zeros consecutivos, e considere F como o evento em que o primeiro bit de uma cadeia de extensão quatro é um 0. A probabilidade de uma cadeia de bits de extensão quatro ter pelo menos dois zeros consecutivos, dado que o primeiro bit da cadeia é um 0, é igual a

$$p(E \mid F) = \frac{p(E \cap F)}{p(F)}.$$

Como $E \cap F = \{0000, 0001, 0010, 0011, 0100\}$, vemos que $p(E \cap F) = 5/16$. Como há oito cadeias de bits de extensão quatro que começam com 0, temos $p(F) = 8/16 = 1/2$.

Conseqüentemente,

$$p(E \mid F) = \frac{5/16}{1/2} = \frac{5}{8}.$$ ◀

EXEMPLO 4 Qual é a probabilidade condicional de uma família com dois filhos ter dois garotos, dado que eles têm pelo menos um garoto? Assuma que cada uma das possibilidades *BB*, *BG*, *GB* e *GG* possam ocorrer igualmente, em que *B* representa um garoto e *G* representa uma garota. (Note que *BG* representa uma família com um garoto mais velho que a garota, enquanto *GB* representa uma família com uma garota mais velha que o garoto.)

Solução: Considere E como o evento em que uma família com duas crianças tem dois garotos, e assuma F como o evento em que uma família com dois filhos tem pelo menos um garoto. Temos que $E = \{BB\}$, $F = \{BB, BG, GB\}$ e $E \cap F = \{BB\}$. Como as quatro possibilidades podem ocorrer igualmente, temos que $p(F) = 3/4$ e $p(E \cap F) = 1/4$. Concluímos que

$$p(E \mid F) = \frac{(E \cap F)}{p(F)} = \frac{1/4}{3/4} = \frac{1}{3}.$$ ◀

Independência

Links Suponha que uma moeda seja lançada três vezes, como descrito na introdução de nossa discussão sobre probabilidade condicional. Saber que a primeira moeda lançada é coroa (evento F) altera a probabilidade de a coroa sair um número ímpar de vezes (evento E)? Em outras palavras, é o caso de $p(E \mid F) = p(E)$? Essa equação é válida para os eventos E e F, pois $p(E \mid F) = 1/2$ e $p(E) = 1/2$. Como essa equação é válida, dizemos que E e F são **eventos independentes.** Quando dois eventos são independentes, a ocorrência de um dos eventos não passa nenhuma informação sobre a probabilidade de o outro evento acontecer.

Como $p(E \mid F) = p(E \cap F)/p(F)$, perguntar se $p(E \mid F) = p(E)$ é o mesmo que perguntar se $p(E \cap F) = p(E)p(F)$. Isso nos leva à Definição 4.

DEFINIÇÃO 4 Os eventos E e F são *independentes* se e somente se $p(E \cap F) = p(E)p(F)$.

EXEMPLO 5 Suponha que E seja o evento que é gerado aleatoriamente por cadeia de bits de extensão quatro que começa com um 1 e F seja o evento em que essa cadeia tenha um número par de 1s. E e F são independentes, se as 16 cadeias de bits de extensão quatro forem igualmente possíveis?

Exemplos Extras

Solução: Há oito cadeias de bits de extensão quatro que começam com um 1: 1000, 1001, 1010, 1011, 1100, 1101, 1110 e 1111. Há também oito cadeias de bits de extensão quatro que têm um número par de 1s: 0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111. Como há 16 cadeias de bits de extensão quatro, temos que

$$p(E) = p(F) = 8/16 = 1/2.$$

Como $E \cap F = \{1111, 1100, 1010, 1001\}$, vemos que

$$p(E \cap F) = 4/16 = 1/4.$$

Como

$$p(E \cap F) = 1/4 = (1/2)(1/2) = p(E)p(F),$$

concluímos que E e F são independentes. ◀

A probabilidade tem muitas aplicações na genética, como mostram os exemplos 6 e 7.

EXEMPLO 6 Assuma, como no Exemplo 4, que cada uma das quatro maneiras de uma família ter duas crianças é igualmente possível. Os eventos E, de uma família com duas crianças ter dois garotos, e F, de uma família com duas crianças ter pelo menos um garoto, são independentes?

Solução: Como $E = \{BB\}$, temos $p(E) = 1/4$. No Exemplo 4, mostramos que $p(F) = 3/4$ e que $p(E \cap F) = 1/4$. Mas $p(E)p(F) = \frac{1}{4} \cdot \frac{3}{4} = \frac{3}{16}$. Portanto, $p(E \cap F) \neq p(E)p(F)$, então, os eventos E e F não são independentes. ◀

EXEMPLO 7 Os eventos E, de uma família com três crianças ter filhos de ambos os sexos, e F, de essa família ter no máximo um garoto, são independentes? Assuma que as oito maneiras de uma família ter as três crianças são igualmente possíveis.

Solução: Pela hipótese, cada uma das oito maneiras de uma família ter três filhos, *BBB*, *BBG*, *BGB*, *BGG*, *GBB*, *GBG*, *GGB* e *GGG*, tem uma probabilidade de $1/8$. Como $E = \{BBG, BGB, BGG, GBB, GBG, GGB\}$, $F = \{BGG, GBG, GGB, GGG\}$ e $E \cap F = \{BGG, GBG, GGB\}$, temos que $p(E) = 6/8 = 3/4$, $p(F) = 4/8 = 1/2$ e $p(E \cap F) = 3/8$. Como

$$p(E)p(F) = \frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8},$$

temos que $p(E \cap F) = p(E)p(F)$, então E e F são independentes. (Esta conclusão pode parecer surpreendente. De fato, se mudarmos o número de crianças, a conclusão pode não ser mais válida. Veja o Exercício 27 no final desta seção.) ◀

Testes de Bernoulli e a Distribuição Binomial

Links

Suponha que um experimento possa ter apenas dois resultados possíveis. Por exemplo, quando um bit é construído aleatoriamente, os resultados possíveis são 0 e 1. Quando uma moeda é jogada, os resultados possíveis são cara e coroa. Cada realização de um experimento com dois resultados possíveis é chamada de **teste de Bernoulli,** depois que James Bernoulli fez importantes contribuições à teoria da probabilidade. Em geral, um resultado possível do teste de Bernoulli é chamado de **sucesso** ou **fracasso.** Se p for a probabilidade de sucesso e q for a probabilidade de fracasso, temos que $p + q = 1$.

Muitos problemas podem ser resolvidos determinando a probabilidade de k sucessos quando um experimento consistir em n testes de Bernoulli mutuamente independentes. (Os testes de Bernoulli são **mutuamente independentes** se a probabilidade condicional de sucesso de qualquer teste dado for p, dada qualquer informação sobre os resultados de outros testes.) Considere o Exemplo 8.

EXEMPLO 8 Uma moeda é modificada para que a probabilidade de sair cara seja $2/3$. Qual é a probabilidade de saírem quatro caras quando a moeda for jogada sete vezes, assumindo que as jogadas são independentes?

Solução: Há $2^7 = 128$ resultados possíveis quando uma moeda é lançada sete vezes. O número de maneiras de que quatro dos sete lançamentos sejam caras é $C(7, 4)$. Como os sete lançamentos

são independentes, a probabilidade de cada um desses resultados (quatro caras e três coroas) é $(2/3)^4(1/3)^3$. Conseqüentemente, a probabilidade de que quatro caras apareçam é

$$C(7, 4)(2/3)^4(1/3)^3 = \frac{35 \cdot 16}{3^7} = \frac{560}{2187}.$$ ◀

Seguindo o mesmo raciocínio como o usado no Exemplo 8, podemos encontrar a probabilidade de k sucessos em n testes de Bernoulli independentes.

TEOREMA 2

A probabilidade de k sucessos em n testes de Bernoulli independentes, com probabilidade de sucesso p e probabilidade de fracasso $q = 1 - p$, é

$$C(n, k)p^k q^{n-k}.$$

Demonstração: Quando n testes de Bernoulli são realizados, o resultado é uma n-upla $(t_1, t_2, \ldots, t_n)$, em que $t_i = S$ (para sucesso) ou $t_i = F$ (para fracasso) para $i = 1, 2, \ldots, n$. Como os n testes são independentes, a probabilidade de cada resultado de n testes ser de k sucessos e $n - k$ fracassos (em qualquer ordem) é $p^k q^{n-k}$. Como há $C(n, k)$ n-uplas de S's e F's que contêm k S's, a probabilidade de k sucessos é

$$C(n, k)p^k q^{n-k}.$$ ◁

Indicamos por $b(k; n, p)$ a probabilidade de k sucessos em n testes de Bernoulli independentes com probabilidade de sucesso p e probabilidade de fracasso $q = 1 - p$. Considerada uma função de k, chamamos essa função de **distribuição binomial.** O Teorema 2 nos diz que $b(k; n, p) = C(n, k)p^k q^{n-k}$.

EXEMPLO 9

Exemplos Extras

Suponha que a probabilidade de um bit 0 ser gerado é 0,9, que a probabilidade de um bit 1 ser gerado é 0,1 e que os bits são gerados independentemente. Qual é a probabilidade de oito bits 0 serem produzidos quando 10 bits forem produzidos?

Solução: Pelo Teorema 2, a probabilidade de oito bits 0 serem produzidos é

$$b(8; 10, 0{,}9) = C(10, 8)(0{,}9)^8(0{,}1)^2 = 0{,}1937102445.$$ ◀

Links

JAMES BERNOULLI (1654–1705) James Bernoulli (também conhecido como Jacob I), nasceu em Basel, na Suíça. Ele é um dos oito matemáticos proeminentes da família Bernoulli (veja na Seção 10.1 a árvore genealógica de matemáticos dos Bernoulli). Seguindo o desejo de seu pai, James estudou teologia e entrou para o seminário. Contudo, contrário aos desejos de seus pais, ele também estudou matemática e astronomia. Viajou pela Europa de 1676 a 1682, aprendendo sobre as últimas descobertas em matemática e nas ciências. Depois de retornar a Basel, em 1682, fundou uma escola de matemática e ciências. Ele foi indicado como professor de matemática para a Universidade de Basel em 1687, permanecendo nesse cargo pelo resto de sua vida.

James Bernoulli é muito conhecido por seu trabalho *Ars Conjectandi*, publicado oito anos depois de sua morte. Nesse trabalho, ele descreve os resultados conhecidos em teoria da probabilidade e em enumeração, normalmente fornecendo demonstrações alternativas dos resultados conhecidos. Seu trabalho também inclui a aplicação da teoria da probabilidade em jogos de azar e sua introdução ao teorema conhecido como **lei dos grandes números.** Esta lei estabelece que se $\epsilon > 0$, como n se torna arbitrariamente grande, a probabilidade aproxima-se de 1, de modo que a fração de vezes de um evento E ocorrer durante n testes é menor que ϵ de $p(E)$.

Note que a soma das probabilidades de haver k sucessos quando n testes de Bernoulli independentes são realizados, para $k = 0, 1, 2, \ldots, n$, é igual a

$$\sum_{k=0}^{n} C(n,k)p^k q^{n-k} = (p+q)^n = 1,$$

como deveria ser. A primeira equação nesta seqüência de equações é uma conseqüência do Teorema Binomial (veja a Seção 5.4). A segunda equação segue do fato de $q = 1 - p$.

Variáveis Aleatórias

Muitos problemas são relativos a um valor numérico associado ao resultado de um experimento. Por exemplo, podemos estar interessados no número total de bits 1 em uma cadeia de 10 bits construída aleatoriamente; ou no número de vezes que coroa aparece quando uma moeda é lançada 20 vezes. Para estudar problemas desse tipo, introduzimos o conceito de variável aleatória.

DEFINIÇÃO 5 Uma *variável aleatória* é uma função do espaço amostral de um experimento para o conjunto dos números reais, ou seja, uma variável aleatória atribui um número real para cada resultado possível.

Lembre-se: Note que uma variável aleatória é uma função. Não é uma variável, e não é aleatória!

EXEMPLO 10 Suponha que uma moeda seja lançada três vezes. Considere $X(t)$ como a variável aleatória que é igual ao número de caras que aparecem quando t é um resultado. Então, $X(t)$ assume os seguintes valores:

$$X(HHH) = 3,$$
$$X(HHT) = X(HTH) = X(THH) = 2,$$
$$X(TTH) = X(THT) = X(HTT) = 1,$$
$$X(TTT) = 0.$$

◀

DEFINIÇÃO 6 A *distribuição* de uma variável aleatória X em um espaço amostral S é o conjunto dos pares $(r, p(X = r))$ para todo $r \in X(S)$, em que $p(X = r)$ é a probabilidade de X assumir o valor r. Uma distribuição é geralmente descrita especificando $p(X = r)$ para cada $r \in X(S)$.

EXEMPLO 11 Como cada um dos oito resultados possíveis quando três moedas são lançadas tem probabilidade $1/8$, a distribuição da variável aleatória $X(t)$ no Exemplo 10 é dada por $P(X = 3) = 1/8$, $P(X = 2) = 3/8$, $P(X = 1) = 3/8$, $P(X = 0) = 1/8$. ◀

EXEMPLO 12 Considere X como a soma dos números que aparecem quando um par de dados é lançado. Quais são os valores desta variável aleatória para os 36 resultados possíveis (i, j), em que i e j são os números que aparecem no primeiro e no segundo dado, respectivamente, quando esses dois dados são lançados?

Solução: A variável aleatória X assume os seguintes valores:

$$
\begin{aligned}
&X((1, 1)) = 2,\\
&X((1, 2)) = X((2, 1)) = 3,\\
&X((1, 3)) = X((2, 2)) = X((3, 1)) = 4,\\
&X((1, 4)) = X((2, 3)) = X((3, 2)) = X((4, 1)) = 5,\\
&X((1, 5)) = X((2, 4)) = X((3, 3)) = X((4, 2)) = X((5, 1)) = 6,\\
&X((1, 6)) = X((2, 5)) = X((3, 4)) = X((4, 3)) = X((5, 2)) = X((6, 1)) = 7,\\
&X((2, 6)) = X((3, 5)) = X((4, 4)) = X((5, 3)) = X((6, 2)) = 8,\\
&X((3, 6)) = X((4, 5)) = X((5, 4)) = X((6, 3)) = 9,\\
&X((4, 6)) = X((5, 5)) = X((6, 4)) = 10,\\
&X((5, 6)) = X((6, 5)) = 11,\\
&X((6, 6)) = 12.
\end{aligned}
$$

◀

Continuaremos nosso estudo sobre as variáveis aleatórias na Seção 6.4, na qual mostraremos como elas podem ser usadas em uma variedade de aplicações.

O Paradoxo do Aniversário

Um famoso problema procura pelo menor número de pessoas necessárias em uma sala para que seja provável que pelo menos duas delas tenham nascido no mesmo dia do mesmo ano. A maioria das pessoas pensa que a resposta, que mostraremos no Exemplo 13, é surpreendentemente pequena. Depois de resolvermos este famoso problema, vamos mostrar como um argumento semelhante pode ser adaptado para resolver uma questão sobre funções de hashing.

EXEMPLO 13 **O Paradoxo do Aniversário** Qual é o número mínimo de pessoas que é necessário ter em uma sala para que a probabilidade de que pelo menos duas delas tenham a mesma data de aniversário seja maior que $1/2$?

Links

Solução: Primeiro estabelecemos algumas hipóteses. Assumimos que os aniversários das pessoas na sala sejam independentes. Além disso, assumimos que cada aniversário seja igualmente provável e que há 366 dias em um ano. (Na realidade, mais pessoas nascem em alguns dias do ano do que em outros, tais como nove meses depois de alguns feriados, incluindo véspera do ano-novo, e apenas anos bissextos têm 366 dias.)

Para encontrar a probabilidade de pelo menos duas das n pessoas na sala terem a mesma data de aniversário, primeiro calculamos a probabilidade p_n de essas pessoas fazerem aniversário em datas diferentes. Então, a probabilidade de pelo menos duas pessoas fazerem aniversário no mesmo dia é $1 - p_n$. Para computar p_n, consideramos as datas de aniversário das n pessoas em ordem fixa. Imagine-as entrando na sala uma de cada vez; vamos computar a probabilidade de cada pessoa que entrar na sala sucessivamente ter uma data de aniversário diferente daquelas que já estão na sala.

O aniversário da primeira pessoa certamente não coincidirá com o aniversário de alguém que já está na sala. A probabilidade de o aniversário da segunda pessoa ser diferente daquele da primeira pessoa é $365/366$, pois a segunda pessoa tem uma data de aniversário diferente quando ele(a) nasceu em um dos 365 dias diferentes do dia em que a primeira pessoa nasceu. (A hipótese de que seja igualmente provável para alguém nascer em um dos 366 dias do ano coloca isso em passos subseqüentes.)

A probabilidade de que a terceira pessoa faça aniversário em uma data diferente daquela da primeira e da segunda pessoa, dado que estas duas pessoas têm datas diferentes de aniversário, é $364/366$. Em geral, a probabilidade da j-ésima pessoa, com $2 \leq j \leq 366$, fazer aniversário em

uma data diferente daquela das $j-1$ pessoas que já estão na sala, dado que estas $j-1$ pessoas têm datas de aniversário diferentes, é

$$\frac{366-(j-1)}{366}=\frac{367-j}{366}.$$

Como assumimos que os aniversários das pessoas na sala são independentes, podemos concluir que a probabilidade de n pessoas na sala fazerem aniversário em dias diferentes é

$$p_n=\frac{365}{366}\frac{364}{366}\frac{363}{366}\cdots\frac{367-n}{366}.$$

Temos que a probabilidade de que entre n pessoas existam pelo menos duas pessoas com a mesma data de aniversário é

$$1-p_n=1-\frac{365}{366}\frac{364}{366}\frac{363}{366}\cdots\frac{367-n}{366}.$$

Para determinar o menor número de pessoas na sala para que a probabilidade de ter pelo menos duas pessoas com a mesma data de aniversário seja maior que $1/2$, usamos a fórmula que encontramos para $1-p_n$ *para* computá-la em ordem crescente de valores de n até ela se tornar maior que $1/2$. (Há aproximações mais sofisticadas que usam cálculo que pode eliminar esta computação, mas não vamos usá-las aqui.) Depois de computação considerável, encontramos que para $n=22$, $1-p_n\approx 0{,}475$, enquanto que para $n=23$, $1-p_n\approx 0{,}506$. Conseqüentemente, o valor mínimo de pessoas necessário para que a probabilidade de pelo menos duas pessoas terem a mesma data de aniversário seja maior que $1/2$ é 23. ◀

A solução para o paradoxo do aniversário leva à solução da questão do Exemplo 14 sobre funções de hashing.

EXEMPLO 14 **Probabilidade de uma Colisão nas Funções de Hashing** Lembre-se da Seção 3.4, que uma função de hashing $h(k)$ é um mapeamento das chaves (dos arquivos que foram guardados na base de dados) para armazenamento de localizações. As funções de hashing mapeiam um grande número de chaves (são aproximadamente 300 milhões de números de Seguro Social nos Estados Unidos) para um conjunto muito menor de armazenamento de localizações. Uma boa função de hashing causa pequenas **colisões,** que são mapeamentos de duas chaves diferentes na mesma localização, quando relativamente poucos arquivos estão em jogo em uma aplicação. Qual é a probabilidade de duas chaves não serem mapeadas na mesma localização por uma função de hashing, ou, em outras palavras, que não haja colisões?

Solução: Para calcular essa probabilidade, assumimos que a probabilidade de uma chave selecionada aleatoriamente ser mapeada para uma localização é $1/m$, em que m é o número de localizações disponíveis, ou seja, a função de hashing distribui chaves uniformemente. (Na prática, as funções de hashing não satisfazem esta hipótese. Entretanto, para uma boa função de hashing, esta hipótese está próxima de estar correta.) Além disso, assumimos que as chaves dos registros selecionados têm probabilidades iguais de serem quaisquer elementos do universo de chaves e que tais chaves são selecionadas independentemente.

Suponha que as chaves sejam $k_1, k_2, \ldots, k_n$. Quando adicionamos o segundo registro, a probabilidade de que ele seja mapeado em uma localização diferente da localização do primeiro registro, que $h(k_2)\neq h(k_1)$, é $(m-1)/m$, porque há $m-1$ localizações livres depois que o primeiro registro foi adicionado. A probabilidade de o terceiro registro ser mapeado para uma localização livre depois que o primeiro e o segundo registro foram colocados sem colisão é $(m-2)/m$. Em geral, a probabilidade do j-ésimo registro ser mapeado para uma localização livre depois que os primeiros $j-1$ registros foram mapeados para as localizações $h(k_1)$, $h(k_2)$, $\ldots$, $h(k_{j-1})$ sem colisão, é $(m-(j-1))/m$, pois $j-1$ das m localizações foram utilizadas.

Como as chaves são independentes, a probabilidade de que todas as n chaves sejam mapeadas em localizações diferentes é

$$p_n = \frac{m-1}{m} \cdot \frac{m-2}{m} \cdot \cdots \cdot \frac{m-n+1}{m}.$$

Daqui segue que a probabilidade de haver pelo menos uma colisão, ou seja, de pelo menos duas chaves serem mapeadas na mesma localização, é

$$1 - p_n = 1 - \frac{m-1}{m} \cdot \frac{m-2}{m} \cdot \cdots \cdot \frac{m-n+1}{m}.$$

As técnicas de cálculo podem ser usadas para encontrar o menor valor de n dado um valor de m, tal que a probabilidade de uma colisão seja maior que um determinado valor. Podemos mostrar que o menor número inteiro n, para que a probabilidade de uma colisão seja maior que $1/2$, é aproximadamente $n = 1{,}177\sqrt{m}$. Por exemplo, quando $m = 1\,000\,000$, o menor número inteiro n, para que a probabilidade de uma colisão seja maior que $1/2$, é 1178. ◀

Algoritmos Monte Carlo

Os algoritmos discutidos até aqui neste livro são todos determinísticos, ou seja, cada algoritmo sempre trabalha da mesma maneira sempre que é dada a mesma entrada. Entretanto, há muitas situações em que precisamos de um algoritmo que faça uma escolha aleatória em um ou mais passos. Quando uma situação dessa aparece, um algoritmo determinístico teria que utilizar um número muito grande, ou até mesmo um número desconhecido, de casos possíveis. Os algoritmos que fazem escolhas aleatórias em um ou mais passos são chamados de **algoritmos probabilísticos.** Discutiremos uma classe particular de algoritmos probabilísticos nesta seção, ou seja, os **algoritmos Monte Carlo,** para problemas de decisões. Os algoritmos Monte Carlo sempre produzem respostas aos problemas, mas resta uma pequena probabilidade de estas respostas serem incorretas. Entretanto, a probabilidade de que uma resposta esteja incorreta diminui rapidamente quando o algoritmo realiza um número suficiente de computações. Os problemas que envolvem decisões têm como resposta "verdadeiro" ou "falso". A designação "Monte Carlo" é uma referência ao famoso cassino em Mônaco; o uso da aleatoriedade e do processo de repetição nesses algoritmos torna-os semelhantes às apostas. Esse nome foi introduzido pelos inventores dos métodos de Monte Carlo, incluindo Stan Ulam, Enrico Fermi e John von Neumann.

Um algoritmo Monte Carlo para problemas que envolvem decisão usa uma seqüência de testes. A probabilidade de o algoritmo responder ao problema corretamente aumenta quanto mais testes forem realizados. Em cada passo do algoritmo, as respostas possíveis são "verdadeiro", que significa que a resposta é "verdadeira" e não é mais necessária nenhuma iteração, ou "desconhecido", que significa que a resposta pode ser "verdadeira" ou "falsa". Depois de executar todas as iterações em tal algoritmo, a resposta final produzida é "verdadeiro" se pelo menos uma iteração fornecer a resposta "verdadeira", e a resposta é "falso" se toda iteração fornecer a resposta "desconhecido". Se a resposta correta for "falso", então o algoritmo responde "falso", porque toda iteração vai fornecer "desconhecido". Entretanto, se a resposta correta for "verdadeiro", então o algoritmo responderá "verdadeiro" ou "falso", porque pode ser possível que cada iteração tenha produzido a resposta "desconhecido" mesmo se a resposta correta for "verdadeiro". Mostraremos que esta possibilidade se torna extremamente improvável quando o número de testes aumenta.

Suponha que p seja a probabilidade de a resposta de um teste ser "verdadeiro", dado que a resposta é "verdadeira". Temos que $1 - p$ é a probabilidade de a resposta ser "desconhecido", dado que a resposta seja "verdadeira". Como o algoritmo responde "falso" quando todas as n iterações fornecem a resposta "desconhecido" e as iterações são realizadas independentemente, a probabilidade de erro é $(1 - p)^n$. Quando $p \neq 0$, essa probabilidade tende a 0 com o número de testes aumentado. Conseqüentemente, a probabilidade de o algoritmo responder "verdadeiro" quando a resposta for "verdadeira" tende a 1.

EXEMPLO 15 **Controle de Qualidade** (Este exemplo é adaptado de [AhUl95].) Suponha que uma fábrica peça chips de processadores em lotes de tamanho n, em que n é um número inteiro positivo. O fabricante do chip testou apenas alguns desses lotes para ter certeza de que todos os chips no lote estavam bons (substituindo qualquer chip defeituoso encontrado durante os testes). Em lotes não testados previamente, a probabilidade de um determinado chip ser ruim foi observada como sendo 0,1 quando um teste aleatório é feito. O fabricante de PCs quer saber se todos os chips do lote são bons. Para fazer isso, o fabricante de PCs pode testar cada chip em um lote para ver se ele é bom. Entretanto, isto requer n testes. Assumindo que cada teste pode ser realizado em um tempo constante, esses testes necessitam de $O(n)$ segundos. O fabricante de PCs pode testar se um lote de chips foi testado pelo fabricante usando menos tempo?

Solução: Podemos usar um algoritmo Monte Carlo para determinar se um lote de chips foi testado pelo fabricante contanto que aceitemos alguma probabilidade de erro. O algoritmo é um programa para responder a pergunta: "Este lote de chips não foi testado pelo fabricante?" Ele trabalha selecionando chips sucessivamente de maneira aleatória a partir do lote e os testa um por um. Quando um chip ruim for encontrado, o algoritmo responde "verdadeiro" e pára. Se um chip testado for bom, o algoritmo responde "desconhecido" e continua com mais uma iteração. Depois que o algoritmo testou um determinado número de chips, digamos k chips, sem conseguir uma resposta "verdadeiro", o algoritmo finaliza com a resposta "falso"; ou seja, o algoritmo conclui que o lote é bom, ou seja, que o fabricante testou todos os chips do lote.

A única maneira de este algoritmo responder incorretamente é concluir que um lote de chips não testados foi testado pelo fabricante. A probabilidade de um chip ser bom, mas vir de um lote não testado, é $1 - 0,1 = 0,9$. Como os eventos de testar chips diferentes de um lote são independentes, a probabilidade de todos os k passos do algoritmo produzir a resposta "desconhecido", dado que o lote de chips não foi testado, é $0,9^k$.

Tomando k suficientemente grande, podemos tornar esta probabilidade tão pequena quanto quisermos. Por exemplo, testando 66 chips, a probabilidade de o algoritmo decidir que um lote foi testado pelo fabricante é $0,9^{66}$, que é menor que 0,001. Ou seja, a probabilidade é menor que 1 em 1000 de o algoritmo responder incorretamente. Note que essa probabilidade é independente de n, o número de chips em um lote, ou seja, o algoritmo Monte Carlo usa um número constante, ou $O(1)$, de testes e necessita de $O(1)$ segundos, não importa quantos chips tem em um lote. À medida que o fabricante de PCs possa viver com uma margem de erro de menos de 1 em 1000, o algoritmo Monte Carlo vai poupar a fábrica de PCs de muitos testes. Se uma margem de erro menor é necessária, o fabricante de PCs pode testar mais chips em cada lote; o leitor pode verificar que 132 testes podem ser feitos para que a margem de erro seja menos de 1 em 1 000 000. ◀

EXEMPLO 16 **Teste Probabilístico para Verificação se um Número É Primo** No Capítulo 3 vimos que um número inteiro composto, ou seja, um número inteiro que não é primo, passa pelo teste de Miller (veja o preâmbulo do Exercício 30 da Seção 3.7) para inteiros menores que $n/4$ base b com $1 < b < n$. Esta observação é a base para um algoritmo Monte Carlo determinar se um número inteiro positivo é primo. Como números primos grandes desenvolvem um papel importante na criptografia de chave pública (veja a Seção 3.7), ser capaz de construir números primos grandes rapidamente se tornou extremamente importante.

O objetivo do algoritmo é responder a questão "n é composto?" Dado um número inteiro n, selecionamos um inteiro b de maneira aleatória com $1 < b < n$ e determinamos se n passa pelo teste de Miller na base b. Se n falhar no teste, a resposta é "verdadeiro" porque n deve ser composto e o algoritmo termina. Por outro lado, repetimos o teste k vezes, em que k é um número inteiro. Cada vez selecionamos um número inteiro b aleatoriamente e determinamos se n passa no teste de Miller na base b. Se a resposta for "desconhecido" em cada passo, o algoritmo responde "falso", ou seja, ele diz que n não é composto, assim, ele é primo. A única possibilidade de o algoritmo dar uma resposta incorreta ocorre quando n é composto e a resposta "desconhecido" é a saída da etapa de uma das k iterações. A probabilidade de um número inteiro composto n passar no teste de Miller para uma seleção aleatória de base b é menor que $1/4$. Como o número inteiro

b com $1 < b < n$ é selecionado de maneira aleatória em cada repetição e estas iterações são independentes, a probabilidade de n ser composto e o algoritmo responder que n é primo é menor que $(1/4)^k$. Tomando k suficientemente grande, podemos tornar essa probabilidade extremamente pequena. Por exemplo, com 10 iterações, a probabilidade de o algoritmo decidir que n é primo quando ele, na verdade, é composto é menor que 1 em 1 000 000. Com 30 iterações, essa probabilidade cai para menos de 1 em 10^{18}, um evento extremamente improvável.

Para construir números primos maiores, digamos com 200 dígitos, podemos escolher aleatoriamente um número inteiro n com 200 dígitos e executar esse algoritmo, com 30 iterações. Se o algoritmo decidir que n é primo, podemos usá-lo como um dos dois primos da chave de codificação do criptosistema RSA. Se n for, na realidade, composto e for usado como parte dessa chave, o procedimento usado para decodificar mensagens não produzirá a mensagem original codificada. A chave é então descartada e dois novos possíveis números primos são usados. ◀

O Método Probabilístico

Discutimos a existência de demonstrações no Capítulo 1 e ilustramos a diferença entre demonstrações de existência construtiva e demonstrações de existência não construtiva. O método probabilístico, introduzido por Paul Erdös e Alfréd Rényi, é uma técnica poderosa que pode ser usada para criar demonstrações de existência não construtiva. Para usar o método probabilístico para demonstrar resultados de um conjunto S, para a existência de um elemento em S com uma determinada propriedade, fixamos probabilidades para os elementos de S. Então, usamos o método a partir da teoria da probabilidade para demonstrar os resultados dos elementos de S. Em particular, podemos mostrar que um elemento com uma determinada propriedade existe, mostrando que a probabilidade de um elemento $x \in S$ ter essa propriedade é positiva. O método probabilístico é baseado na proposição equivalente do Teorema 3.

TEOREMA 3 **O MÉTODO PROBALÍSTICO** Se a probabilidade de um elemento de um conjunto S não ter uma determinada propriedade é menor que 1, existirá um elemento em S com essa propriedade.

Uma demonstração da existência baseada no método probabilístico é não construtiva porque não encontra um determinado elemento com a propriedade desejada.

Ilustramos o poder do método probabilístico encontrando um limite menor para o número de Ramsey $R(k, k)$. Relembre a partir da Seção 5.2 que $R(k, k)$ é igual ao número mínimo de pessoas em uma festa necessário para garantir que há, pelo menos, k amigos mútuos ou k inimigos mútuos (assumindo que duas pessoas são amigas ou inimigas).

TEOREMA 4 Se k é um número inteiro com $k \geq 2$, então $R(k, k) \geq 2^{k/2}$.

Demonstração: Notamos que o teorema é válido para $k = 2$ e $k = 3$, porque $R(2, 2) = 2$ e $R(3, 3) = 6$, como foi mostrado na Seção 5.2. Agora, suponha que $k \geq 4$. Usaremos o método probabilístico para mostrar que se houver menos pessoas que $2^{k/2}$ na festa, é possível que nenhuma k delas sejam amigas ou inimigas mútuas. Isto mostrará que $R(k, k)$ é, pelo menos, $2^{k/2}$.

Para usar o método probabilístico, assumimos que é igualmente provável que duas pessoas sejam amigas ou inimigas. (Note que essa hipótese não tem base real.) Suponha que há n pessoas na festa. Temos que há $\binom{n}{k}$ conjuntos diferentes de k pessoas na festa, que listamos como $S_1, S_2, \ldots, S_{\binom{n}{k}}$. Considere E_i como o evento em que todas as k pessoas em S_i são amigas mútuas

ou inimigas mútuas. A probabilidade de haver k amigos mútuos ou k inimigos mútuos entre as n pessoas é igual a $p(\bigcup_{i=1}^{\binom{n}{k}} E_i)$.

De acordo com a nossa hipótese, é igualmente possível que duas pessoas sejam amigas ou inimigas. A probabilidade de duas pessoas serem amigas é igual à probabilidade de duas serem inimigas; ambas as probabilidades são iguais a $1/2$. Além disso, há $\binom{k}{2} = k(k-1)/2$ pares de pessoas em S_i, pois há k pessoas em S_i. Assim, a probabilidade de todas as k pessoas em S_i serem amigas mútuas e a probabilidade de todas as k pessoas em S_i serem inimigas mútuas é igual a $(1/2)^{k(k-1)/2}$. Daqui segue que $p(E_i) = 2(1/2)^{k(k-1)/2}$.

A probabilidade de haver k amigos mútuos ou k inimigos mútuos no grupo de n pessoas é igual a $p(\bigcup_{i=1}^{\binom{n}{k}} E_i)$. Usando a Inequação de Boole (Exercício 15), temos que

$$p\left(\bigcup_{i=1}^{\binom{n}{k}} E_i\right) \leq \sum_{i=1}^{\binom{n}{k}} p(E_i) = \binom{n}{k} \cdot 2\left(\frac{1}{2}\right)^{k(k-1)/2}.$$

A partir do Exercício 17 da Seção 5.4, temos $\binom{n}{k} \leq n^k/2^{k-1}$. Assim,

$$\binom{n}{k} 2\left(\frac{1}{2}\right)^{k(k-1)/2} \leq \frac{n^k}{2^{k-1}} 2\left(\frac{1}{2}\right)^{k(k-1)/2}.$$

Agora, se $n < 2^{k/2}$, temos

$$\frac{n^k}{2^{k-1}} 2\left(\frac{1}{2}\right)^{k(k-1)/2} < \frac{2^{k(k/2)}}{2^{k-1}} 2\left(\frac{1}{2}\right)^{k(k-1)/2} = 2^{2-(k/2)} \leq 1$$

em que a última etapa é válida porque $k \geq 4$.

Podemos concluir que $p(\bigcup_{i=1}^{\binom{n}{k}} E_i) < 1$ quando $k \geq 4$. Assim, a probabilidade do evento complementar, de que não há conjunto de k amigos mútuos ou inimigos mútuos na festa, é maior que 0. Temos que se $n < 2^{k/2}$, então há, pelo menos, um conjunto que não tem nenhum subconjunto de k pessoas que sejam amigas ou inimigas mútuas. ◁

Exercícios

1. Qual probabilidade deve ser designada para o resultado de cara quando uma moeda viciada é lançada, se cara é três vezes mais provável de sair que coroa? Qual probabilidade deve ser designada para o resultado de coroa?
2. Encontre a probabilidade de cada um dos resultados quando um dado viciado é lançado, se um 3 é duas vezes mais provável de aparecer que os outros cinco números do dado.
3. Encontre a probabilidade de cada um dos resultados quando um dado viciado é lançado, se os números 2 e 4 têm três vezes mais chance de aparecer que os outros quatro números do dado e é igualmente possível sair um 2 ou um 4.
4. Mostre que as condições (*i*) e (*ii*) são válidas pela definição de probabilidade de Laplace, quando os resultados são igualmente possíveis.
5. Um par de dados é viciado. A probabilidade de aparecer um 4 no primeiro dado é $2/7$ e a probabilidade de aparecer um 3 no segundo dado é $2/7$. Outros resultados para cada dado aparecem com a probabilidade de $1/7$. Qual é a probabilidade de aparecer um 7 como a soma dos números quando os dois dados são lançados?
6. Qual é a probabilidade destes eventos quando selecionamos aleatoriamente uma permutação de $\{1, 2, 3\}$?
   a) 1 precede 3.
   b) 3 precede 1.
   c) 3 precede 1 e 3 precede 2.
7. Qual é a probabilidade destes eventos quando selecionamos aleatoriamente uma permutação de $\{1, 2, 3, 4\}$?
   a) 1 precede 4.
   b) 4 precede 1.
   c) 4 precede 1 e 4 precede 2.
   d) 4 precede 1, 4 precede 2 e 4 precede 3.
   e) 4 precede 3 e 2 precede 1.

8. Qual é a probabilidade destes eventos quando selecionamos aleatoriamente uma permutação de $\{1, 2, \ldots, n\}$, em que $n \geq 4$?
- **a)** 1 precede 2.
- **b)** 2 precede 1.
- **c)** 1 precede imediatamente 2.
- **d)** n precede 1 e $n-1$ precede 2.
- **e)** n precede 1 e n precede 2.

9. Qual é a probabilidade destes eventos quando selecionamos aleatoriamente uma permutação das 26 letras minúsculas do alfabeto inglês?
- **a)** A permutação consiste nas letras em ordem alfabética inversa.
- **b)** z é a primeira letra da permutação.
- **c)** z precede a na permutação.
- **d)** a precede imediatamente z na permutação.
- **e)** a precede imediatamente m, que precede imediatamente z na permutação.
- **f)** m, n e o estão nos seus lugares originais na permutação.

10. Qual é a probabilidade destes eventos quando selecionamos aleatoriamente uma permutação das 26 letras minúsculas do alfabeto inglês?
- **a)** As primeiras 13 letras da permutação estão em ordem alfabética.
- **b)** a é a primeira letra da permutação e z é a última letra.
- **c)** a e z estão juntas na permutação.
- **d)** a e b não estão juntas na permutação.
- **e)** a e z estão separadas por, no mínimo, 23 letras na permutação.
- **f)** z precede a e b na permutação.

11. Suponha que E e F são eventos, tal que $p(E) = 0{,}7$ e $p(F) = 0{,}5$. Mostre que $p(E \cup F) \geq 0{,}7$ e $p(E \cap F) \geq 0{,}2$.

12. Suponha que E e F são eventos, tal que $p(E) = 0{,}8$ e $p(F) = 0{,}6$. Mostre que $p(E \cup F) \geq 0{,}8$ e $p(E \cap F) \geq 0{,}4$.

13. Mostre que se E e F são eventos, então $p(E \cap F) \geq p(E) + p(F) - 1$. Isto é conhecido como **Inequação de Bonferroni.**

14. Use a indução matemática para demonstrar a seguinte generalização da Inequação de Bonferroni:

$$p(E_1 \cap E_2 \cap \cdots \cap E_n) \geq p(E_1) + p(E_2) + \cdots + p(E_n) - (n-1),$$

em que $E_1, E_2, \ldots, E_n$ são n eventos.

☞ **15.** Mostre que se $E_1, E_2, \ldots, E_n$ são eventos de um espaço amostral finito, então

$$p(E_1 \cup E_2 \cup \cdots \cup E_n) \leq p(E_1) + p(E_2) + \cdots + p(E_n).$$

Isto é conhecido como **Inequação de Boole.**

16. Mostre que se E e F são eventos independentes, então $\overline{E}$ e $\overline{F}$ também são eventos independentes.

17. Se E e F são eventos independentes, prove ou negue que $\overline{E}$ e F são necessariamente eventos independentes.

Nos exercícios 18, 20 e 21 assuma que o ano tem 366 dias e todos os aniversários são igualmente possíveis. No Exercício 19 assuma que é igualmente possível uma pessoa nascer em um dado mês do ano.

18. **a)** Qual é a probabilidade de duas pessoas escolhidas aleatoriamente terem nascido no mesmo dia da semana?
- **b)** Qual é a probabilidade de, em um grupo de n pessoas escolhidas aleatoriamente, haver pelo menos duas que nasceram no mesmo dia da semana?
- **c)** Quantas pessoas escolhidas aleatoriamente são necessárias para que a probabilidade seja maior que $1/2$ de haver pelo menos duas pessoas nascidas no mesmo dia da semana?

19. **a)** Qual é a probabilidade de duas pessoas escolhidas aleatoriamente terem nascido durante o mesmo mês do ano?
- **b)** Qual é a probabilidade de, em um grupo de n pessoas escolhidas aleatoriamente, haver pelo menos duas que nasceram no mesmo mês do ano?
- **c)** Quantas pessoas escolhidas aleatoriamente são necessárias para que a probabilidade seja maior que $1/2$ de haver pelo menos duas pessoas nascidas no mesmo mês do ano?

20. Encontre o menor número de pessoas necessárias em uma escolha aleatória para que a probabilidade de pelo menos uma delas fazer aniversário hoje ultrapasse $1/2$.

21. Encontre o menor número de pessoas necessárias em uma escolha aleatória para que a probabilidade de pelo menos duas delas terem nascido em 1º de abril ultrapasse $1/2$.

***22.** O dia 29 de fevereiro ocorre em anos bissextos. Os anos divisíveis por 4, mas não por 100, são todos anos bissextos. Anos divisíveis por 100, mas não por 400, não são anos bissextos, mas os anos divisíveis por 400 são bissextos.
- **a)** Qual a probabilidade de distribuição dos aniversários que deve ser usada para refletir a freqüência com que o dia 29 de fevereiro ocorre?
- **b)** Usando a probabilidade de distribuição do item (a), qual é a probabilidade de, em um grupo de n pessoas, pelo menos duas terem a mesma data de aniversário?

23. Qual é a probabilidade condicional de quatro caras aparecerem quando uma moeda não-viciada é lançada cinco vezes, dado que, na primeira vez, vira uma cara?

24. Qual é a probabilidade condicional de quatro caras aparecerem quando uma moeda não-viciada é lançada cinco vezes, dado que, na primeira vez, vira uma coroa?

25. Qual é a probabilidade condicional de uma cadeia de bits de extensão quatro construída aleatoriamente conter pelo menos dois bits 0 consecutivos, dado que o primeiro bit é 1? (Assuma que as probabilidades de 0 e 1 são as mesmas.)

26. Considere E como o evento de uma cadeia de bits de extensão três construída aleatoriamente que tenha um número ímpar de bits 1 e considere F como o evento em que a cadeia começa com 1. E e F são independentes?

27. Considere E e F como os eventos de uma família de n crianças ter filhos de ambos os sexos e ter, no máximo, um garoto, respectivamente. E e F são independentes se
a) $n = 2$? **b)** $n = 4$? **c)** $n = 5$?

28. Assuma que a probabilidade de uma criança ser garoto é 0,51 e que os sexos das crianças nascidas em uma família são independentes. Qual é a probabilidade de uma família com cinco crianças ter

a) três garotos?

b) pelo menos um garoto?

c) pelo menos uma garota?

d) todas as crianças do mesmo sexo?

29. Um grupo de seis pessoas participa do jogo "cara ou coroa" para determinar quem comprará os refrigerantes. Cada pessoa lança uma moeda. Se houver uma pessoa com o resultado diferente de qualquer outra do grupo, essa pessoa tem de comprar os refrigerantes. Qual é a probabilidade de haver uma pessoa fora depois de as moedas serem jogadas uma vez?

30. Encontre a probabilidade de uma cadeia de bits de extensão 10 construída aleatoriamente não ter um 0, se os bits são independentes e se

a) um bit 0 e um bit 1 são igualmente possíveis.

b) a probabilidade de um bit ser 1 é 0,6.

c) a probabilidade de o i-ésimo bit ser um 1 é $1/2^i$ para $i = 1, 2, 3, \ldots, 10$.

31. Encontre a probabilidade de uma família com cinco crianças não ter um garoto, se os sexos das crianças são independentes e se

a) um garoto e uma menina são igualmente possíveis.

b) a probabilidade de ser um garoto é 0,51.

c) a probabilidade da i-ésima criança ser um garoto é $0{,}51 - (i/100)$.

32. Encontre a probabilidade de uma cadeia de bits de extensão 10 construída aleatoriamente que começa com um 1 ou termina com 00 para as mesmas condições dos itens (a), (b) e (c) do Exercício 30, se os bits são construídos independentemente.

33. Encontre a probabilidade de a primeira criança de uma família com cinco filhos ser um garoto ou de, pelo menos, as últimas duas crianças da família serem meninas, para as mesmas condições dos itens (a), (b) e (c) do Exercício 31.

34. Encontre cada uma das seguintes probabilidades quando n testes de Bernoulli independentes forem realizados com probabilidade de sucesso p.

a) a probabilidade de não ter sucesso.

b) a probabilidade de, pelo menos, um sucesso.

c) a probabilidade de, no máximo, um sucesso.

d) a probabilidade de, pelo menos, dois sucessos.

35. Encontre cada uma das seguintes probabilidades quando n testes de Bernoulli independentes forem realizados com probabilidade de sucesso p.

a) a probabilidade de não ter fracasso.

b) a probabilidade de, pelo menos, um fracasso.

c) a probabilidade de, no máximo, um fracasso.

d) a probabilidade de, pelo menos, dois fracassos.

36. Use a indução matemática para demonstrar que se E_1, E_2, . . . , E_n é uma seqüência de n eventos disjuntos em um espaço amostral S, em que n é um número inteiro positivo, então $p(\bigcup_{i=1}^{n} E_i) = \sum_{i=1}^{n} p(E_i)$.

***37.** (*Requer cálculo*) Mostre que se E_1, E_2, . . . é uma seqüência infinita de eventos disjuntos em um espaço amostral S, então $p(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} p(E_i)$. [*Dica:* Use o Exercício 36 e considere os limites.]

38. Um par de dados é lançado em uma localização remota e quando você pergunta a um observador honesto se pelo menos em um dado saiu seis, esse observador honesto responde afirmativamente.

a) Qual é a probabilidade de a soma dos números que aparecem nos dois dados ser sete, dada a informação fornecida pelo observador honesto?

b) Suponha que o observador nos diga que pelo menos em um dado tenha saído cinco. Qual é a probabilidade de a soma dos números que saíram nos dados ser sete, dada essa informação?

****39.** Este exercício emprega o método probabilístico para demonstrar um resultado sobre torneios Round-robin. Em um **torneio Round-robin** com m jogadores, dois jogadores participam de um jogo no qual um ganha e o outro perde.

Queremos encontrar condições de números inteiros positivos m e k com $k < m$ para que seja possível que os resultados do torneio tenham a propriedade de que em todo conjunto de k jogadores haja um jogador que ganha de todos os outros membros do conjunto. Para que possamos usar o argumento probabilístico para desenvolver conclusões sobre esses torneios, assumimos que quando dois jogadores competem é igualmente possível que um deles vença, e assumimos que os resultados dos diferentes jogos são independentes. Considere E como o evento de que para todo conjunto S com k jogadores, em que k é um número inteiro positivo menor que m, há um jogador que vence todos os k jogadores em S.

a) Mostre que $p(\overline{E}) \leq \sum_{j=1}^{\binom{n}{k}} p(F_j)$, em que F_j é o evento em que nenhum jogador ganha de todos os k jogadores a partir do j-ésimo conjunto em uma lista de $\binom{m}{k}$ conjuntos de k jogadores.

b) Mostre que a probabilidade de F_j é $(1-2^{-k})^{m-k}$.

c) Conclua a partir dos itens (a) e (b) que $p(E) \leq \binom{m}{k}(1 - 2^{-k})^{m-k}$ e, portanto, deve haver um torneio com a propriedade descrita se $\binom{m}{k}(1 - 2^{-k})^{m-k} < 1$.

d) Use o item (c) para encontrar os valores de m para que haja um torneio com m jogadores, tal que em todo conjunto S de dois jogadores haja um jogador que vença ambos os jogadores de S. Repita para o conjunto de três jogadores.

***40.** Desenvolva um algoritmo Monte Carlo que determine se uma permutação dos números inteiros 1 até n já foi ordenado (ou seja, está em ordem crescente) ou, em vez disso, é uma permutação aleatória. Um passo do algoritmo deve responder "verdadeiro" se ele determinar que a lista não está ordenada, e "desconhecido" se ocorrer o contrário. Depois de k passos, o algoritmo decide que os números inteiros estão ordenados se a resposta for "desconhecido" em cada etapa. Mostre que com o aumento do número de passos, a probabilidade de o algoritmo produzir uma resposta incorreta é extremamente pequena. [*Dica:* Para cada passo, teste se certos elementos estão na ordem correta. Tenha certeza de que esses testes são independentes.]

41. Use o pseudocódigo para escrever o teste probabilístico primário descrito no Exemplo 16.

6.3 Teorema de Bayes

Introdução

Existem muitos casos em que queremos saber a probabilidade de um determinado evento ocorrer com base em evidências parciais. Por exemplo, suponha que saibamos a porcentagem de pessoas que tem determinada doença para a qual há um diagnóstico muito acurado. As pessoas que têm diagnóstico positivo para essa doença gostariam de saber a possibilidade de elas realmente terem essa doença. Nesta seção, introduziremos um resultado que pode ser usado para determinar essa probabilidade, ou seja, a probabilidade de uma pessoa ter a doença quando a diagnosticam positivamente. Para usar esse resultado, precisaremos saber a porcentagem de pessoas que não tem a doença, mas que foram diagnosticadas positivamente, e a porcentagem de pessoas que tem a doença, mas que foram diagnosticadas negativamente.

Da mesma forma, suponha que saibamos a porcentagem de e-mails recebidos que é spam. Veremos que podemos determinar a possibilidade de um e-mail recebido ser spam usando a ocorrência de palavras na mensagem. Para determinar essa probabilidade, precisamos saber a porcentagem de e-mails recebidos que é spam, a porcentagem de mensagens que é spam nas quais cada uma das palavras ocorre e a porcentagem de mensagens que não é spam nas quais cada uma das palavras ocorre.

O resultado que podemos usar para resolver questões como as anteriores é chamado de Teorema de Bayes e é datado de antes do século XVIII. Nas duas décadas passadas, o Teorema de Bayes foi muito aplicado para estimar as probabilidades baseadas em evidências parciais em diversas áreas, tais como medicina, direito, inteligência artificial, engenharia e desenvolvimento de softwares.

O Teorema de Bayes

Ilustramos a idéia que está por trás do Teorema de Bayes com um exemplo que mostra que, quando uma informação adicional está disponível, podemos derivar uma estimativa mais real de que um determinado evento ocorra. Ou seja, suponha que saibamos $p(F)$, a probabilidade de um evento F acontecer, mas tenhamos conhecimento de que um evento E ocorre. Então, a probabilidade condicional de que F aconteça, dado que E ocorre, $p(F \mid E)$, é uma estimativa mais real do que $p(F)$, que F aconteça. No Exemplo 1, veremos que podemos encontrar $p(F \mid E)$ quando sabemos $p(F)$, $p(E \mid F)$ e $p(E \mid \overline{F})$.

EXEMPLO 1 Temos duas caixas. A primeira contém duas bolas verdes e sete bolas vermelhas; a segunda contém quatro bolas verdes e três bolas vermelhas. Bob seleciona uma bola escolhendo primeiro uma das duas caixas aleatoriamente. Então, ele seleciona uma das bolas dessa caixa aleatoriamente. Se Bob escolheu uma bola vermelha, qual é a probabilidade de ele ter escolhido uma bola da primeira caixa?

Exemplos Extras

Solução: Considere E como o evento de Bob ter escolhido uma bola vermelha; $\overline{E}$ é o evento de Bob ter escolhido uma bola verde. Considere F como o evento de Bob ter escolhido uma bola da primeira caixa; $\overline{F}$ é o evento de Bob ter escolhido uma bola da segunda caixa. Queremos encontrar $p(F \mid E)$, a probabilidade de que a bola que Bob escolheu foi da primeira caixa, dado que ela é vermelha. Pela definição de probabilidade condicional, temos $p(F \mid E) = p(F \cap E)/p(E)$. Podemos usar a informação fornecida para determinar $p(F \cap E)$ e $p(E)$ para que possamos encontrar $p(F \mid E)$?

Primeiro note que como a primeira caixa contém sete bolas vermelhas de um total de nove bolas, sabemos que $p(E \mid F) = 7/9$. Da mesma forma, como a segunda caixa contém três bolas vermelhas em um total de sete bolas, sabemos que $p(E \mid \overline{F}) = 3/7$. Assumimos que Bob seleciona uma caixa aleatoriamente, assim $p(F) = p(\overline{F}) = 1/2$. Como $p(E \mid F) = p(E \cap F)/p(F)$, temos que $p(E \cap F) = p(E \mid F)p(F) = \frac{7}{9} \cdot \frac{1}{2} = \frac{7}{18}$ [como relembramos anteriormente, está é uma das

quantidades que precisamos encontrar para determinar $p(F \mid E)$]. Da mesma forma, como $p(E \mid \overline{F}) = p(E \cap \overline{F})/p(\overline{F})$, temos que $p(E \cap \overline{F}) = p(E \mid \overline{F})\,p(\overline{F}) = \frac{3}{7} \cdot \frac{1}{2} = \frac{3}{14}$.

Podemos agora encontrar $p(E)$. Note que $E = (E \cap F) \cup (E \cap \overline{F})$, em que $E \cap F$ e $E \cap \overline{F}$ são conjuntos disjuntos. (Se x pertencer a $E \cap F$ e $E \cap \overline{F}$, então x pertence a F e $\overline{F}$, o que é impossível.) Segue-se, então, que

$$p(E) = p(E \cap F) + p(E \cap \overline{F}) = \frac{7}{18} + \frac{3}{14} = \frac{49}{126} + \frac{27}{126} = \frac{76}{264} = \frac{38}{63}.$$

Encontramos que $p(F \cap E) = 7/18$ e $p(E) = 38/63$. Concluímos que

$$p(F \mid E) = \frac{p(F \cap E)}{p(E)} = \frac{7/18}{38/63} = \frac{49}{76} \approx 0{,}645.$$

Antes de termos qualquer informação adicional, assumimos que a probabilidade de Bob escolher a primeira caixa era $1/2$. Entretanto, com a informação adicional de que a bola escolhida aleatoriamente é vermelha, essa probabilidade aumenta em, aproximadamente, 0,645. Ou seja, a probabilidade de Bob ter escolhido uma bola da primeira caixa aumenta de $1/2$, quando nenhuma informação adicional está disponível, para 0,645, uma vez que sabemos que a bola escolhida era vermelha. ◀

Usando o mesmo tipo de argumento do Exemplo 1, podemos encontrar a probabilidade condicional de um evento F acontecer, dado que um evento E aconteceu, quando sabemos que $p(E \mid F)$, $p(E \mid \overline{F})$ e $p(F)$. O resultado que podemos obter é chamado de **Teorema de Bayes**, que recebe esse nome em homenagem a Thomas Bayes, um matemático e pastor britânico do século XVIII que introduziu esse resultado.

TEOREMA 1

TEOREMA DE BAYES Suponha que E e F são eventos de um espaço amostral S, tal que $p(E) \neq 0$ e $p(F) \neq 0$. Então,

$$p(F \mid E) = \frac{p(E \mid F)p(F)}{p(E \mid F)p(F) + p(E \mid \overline{F})p(\overline{F})}.$$

Demonstração: A definição de probabilidade condicional nos diz que $p(F \mid E) = p(E \cap F)/p(E)$ e $p(E \mid F) = p(E \cap F)/p(F)$. Assim, $p(E \cap F) = p(F \mid E)p(E)$ e $p(E \cap F) = p(E \mid F)p(F)$. Igualando as duas expressões por $p(E \cap F)$, mostramos que

$$p(F \mid E)p(E) = p(E \mid F)p(F).$$

Dividindo os dois lados por $p(E)$, encontramos que

$$p(F \mid E) = \frac{p(E \mid F)p(F)}{p(E)}.$$

Para completar a demonstração, mostramos que $p(E) = p(E \mid F)p(F) + p(E \mid \overline{F})p(\overline{F})$. Primeiro, note que $E = E \cap S = E \cap (F \cup \overline{F}) = (E \cap F) \cup (E \cap \overline{F})$. Além disso, $E \cap F$ e $E \cap \overline{F}$ são disjuntos, porque se $x \in E \cap F$ e $x \in E \cap \overline{F}$, então $x \in F \cap \overline{F} = \emptyset$. Conseqüentemente, $p(E) =$

$p(E \cap \overline{F}) + p(E \cap \overline{F})$. Já mostramos que $p(E \cap F) = p(E \mid F)p(F)$. Além disso, temos $p(E \mid \overline{F}) = p(E \cap \overline{F})/p(\overline{F})$, o que mostra que $p(E \cap \overline{F}) = p(E \mid \overline{F})p(\overline{F})$. Daí segue que

$$p(E) = p(E \cap F) + p(E \cap \overline{F}) = p(E \mid F)p(F) + p(E \mid \overline{F})p(\overline{F}).$$

Para completar a demonstração, inserimos esta expressão para $p(E)$ na equação $p(F \mid E) = p(E \mid F)p(F)/p(E)$. E demonstramos que

Links

$$p(F \mid E) = \frac{p(E \mid F)p(F)}{p(E \mid F)p(F) + p(E \mid \overline{F})p(\overline{F})}. \quad \triangleleft$$

O Teorema de Bayes pode ser usado para determinar a probabilidade de alguém que foi diagnosticado positivo para uma doença realmente ter a doença. Os resultados obtidos a partir do Teorema de Bayes são geralmente surpreendentes, como mostra o Exemplo 2.

EXEMPLO 2 Suponha que uma pessoa em 100 000 tenha uma determinada doença rara para a qual há um teste diagnóstico preciso e correto. Esse teste está correto 99% das vezes quando é dado para alguém com a doença e está correto 99,5% das vezes quando é dado para alguém que não tem a doença. Dada essa informação, podemos encontrar

(a) a probabilidade de alguém que foi diagnosticado positivo para a doença ter a doença?
(b) a probabilidade de alguém que foi diagnosticado negativo para a doença não ter a doença?

Alguém que foi diagnosticado positivo deve ficar preocupado com a informação?

Solução: (a) Considere F como o evento de uma pessoa ter a doença e considere E como o evento de que essa pessoa foi diagnosticada positivamente para a doença. Queremos computar $p(F \mid E)$. Para usar o Teorema de Bayes para computar $p(F \mid E)$, precisamos encontrar $p(E \mid F)$, $p(E \mid \overline{F})$, $p(F)$ e $p(\overline{F})$.

Sabemos que um pessoa em 100 000 tem essa doença, assim $p(F) = 1/100\,000 = 0{,}00001$ e $p(\overline{F}) = 1 - 0{,}00001 = 0{,}99999$. Como alguém que tem a doença é diagnosticado positivamente 99% das vezes, sabemos que $p(E \mid F) = 0{,}99$; essa é a probabilidade de um positivo verdadeiro, ou seja, que alguém com a doença seja diagnosticado positivamente. Sabemos também que $p(\overline{E} \mid F) = 0{,}01$; essa é a probabilidade de um negativo falso, ou seja, que alguém que tem a doença seja diagnosticado negativo. Além disso, como alguém que não tem a doença é diagnosticado negativo em 99,5% das vezes, sabemos que $p(\overline{E} \mid \overline{F}) = 0{,}995$. Essa é a probabilidade de um negativo verdadeiro, ou seja, que alguém sem a doença é diagnosticado negativo. Por fim, sabemos que $p(E \mid \overline{F}) = 0{,}005$; essa é a probabilidade de um positivo falso, ou seja, que alguém sem a doença seja diagnosticado positivo.

A probabilidade de alguém ser diagnosticado positivo para a doença e ter a doença realmente é $p(F \mid E)$. Pelo Teorema de Bayes, sabemos que

$$\begin{aligned} p(F \mid E) &= \frac{p(E \mid F)p(F)}{p(E \mid F)p(F) + p(E \mid \overline{F})p(\overline{F})} \\ &= \frac{(0{,}99)(0{,}00001)}{(0{,}99)(0{,}00001) + (0{,}005)(0{,}99999)} \approx 0{,}002. \end{aligned}$$

Isso significa que apenas 0,2% das pessoas que são diagnosticadas positivamente tem realmente a doença. Como a doença é extremamente rara, o número de positivos falsos no diagnóstico é maior que o número de positivos verdadeiros, fazendo com que a porcentagem de pessoas diagnosticadas positivamente que realmente tem a doença seja muito pequena. A pessoa que foi

diagnosticada positivamente não deve se preocupar com o resultado, a não ser que ela realmente tenha a doença.

(b) A probabilidade de alguém que foi diagnosticado negativamente não ter a doença é $p(\overline{E} \mid \overline{F})$. Pelo Teorema de Bayes, sabemos que

$$p(\overline{F} \mid \overline{E}) = \frac{p(\overline{E} \mid \overline{F})p(\overline{F})}{p(\overline{E} \mid \overline{F})p(\overline{F}) + p(\overline{E} \mid F)p(F)}$$
$$= \frac{(0,995)(0,99999)}{(0,995)(0,99999) + (0,01)(0,00001)} \approx 0,9999999.$$

Conseqüentemente, 99,99999% das pessoas que são diagnosticadas negativamente não têm a doença. ◀

Note que na proposição do Teorema de Bayes, os eventos F e $\overline{F}$ são mutuamente exclusivos e cobrem todo o espaço amostral S (ou seja, $F \cup \overline{F} = S$). Podemos estender o Teorema de Bayes a qualquer grupo de eventos mutuamente exclusivos que cobrem todo o espaço amostral S, da maneira a seguir.

TEOREMA 2

GENERALIZAÇÃO DO TEOREMA DE BAYES Suponha que E seja um evento do espaço amostral S e que $F_1, F_2, \ldots, F_n$ sejam eventos mutuamente exclusivos, tal que $\bigcup_{i=1}^{n} F_1 = S$. Assuma que $p(E) \neq 0$ e $p(F_i) \neq 0$ para $i = 1, 2, \ldots, n$. Então,

$$p(F_j \mid E) = \frac{p(E \mid F_j)p(F_j)}{\sum_{i=1}^{n} p(E \mid F_i)p(F_i)}.$$

Deixamos a demonstração dessa versão generalizada do Teorema de Bayes como Exercício 17.

Links

THOMAS BAYES (1702–1761) Bayes era filho de um pastor de uma seita religiosa conhecida como Não-conformistas. Essa seita foi considerada herética no século XVIII na Grã-Bretanha. Como existia um segredo sobre a vida dos Não-conformistas, pouco é conhecido sobre Thomas Bayes. Quando Thomas era jovem, sua família se mudou para Londres. Ele foi educado de forma privada; os filhos de seguidores dos Não-conformistas geralmente não freqüentavam a escola. Em 1719, Bayes entrou para a Universidade de Edinburgh, onde estudou lógica e teologia. Ele foi ordenado como pastor dos Não-conformistas como seu pai, e começou seu trabalho auxiliando-o. Em 1733, tornou-se pastor da Capela Presbiteriana em Tunbridge Wells, sudeste de Londres, onde permaneceu até 1752.

Bayes é conhecido por seu ensaio sobre probabilidade publicado em 1764, três anos depois de sua morte. Esse ensaio foi enviado a Royal Society por um amigo que o encontrou entre os papéis deixados quando Bayes morreu. Na sua introdução, Bayes afirma que seu objetivo era encontrar um método que pudesse medir a probabilidade de um evento acontecer, assumindo que não sabemos nada sobre ele, mas que, considerando as mesmas circunstâncias, aconteceu em alguma proporção de tempo. As conclusões de Bayes foram aceitas pelo grande matemático francês Laplace, mas logo foram questionadas por Boole, em seu livro *Laws of Thought*. Desde então, as técnicas de Bayes têm sido assunto para controvérsias.

Bayes escreveu um artigo que foi também publicado postumamente: "Uma introdução à Doutrina de Fluxão, e uma Defesa dos Matemáticos Contra as Objeções do Autor de O Analista", que apoiava as bases lógicas do cálculo. Bayes foi eleito como um Membro da Royal Society em 1742, com o apoio de membros importantes, mesmo não tendo ainda publicado trabalhos matemáticos. A única publicação conhecida de Bayes durante sua vida foi supostamente um livro místico, intitulado *Divine Benevolence*, que discute a causa original e última proposta do universo. Embora o livro seja comumente atribuído a Bayes, nenhum nome de autor aparece na página de início, e pensa-se ser o trabalho inteiro de procedência duvidosa. As evidências do talento matemático de Bayes vêm de algumas anotações certamente escritas por ele, que contêm muito trabalho matemático, incluindo discussões sobre probabilidade, trigonometria, geometria, soluções de equações, séries e cálculo diferencial. Há também seções sobre filosofia natural, nas quais Bayes examina tópicos como eletricidade, ótica e mecânica celestial. Bayes é autor também de uma publicação matemática sobre séries assintóticas, que apareceu depois de sua morte.

Filtros de Spam de Bayes

Links

A maioria das caixas de entrada eletrônicas recebe uma enxurrada de mensagens indesejadas e não solicitadas, conhecidas como **spam.** Com as ameaças de os spam lotarem as caixas de e-mail, uma grande quantidade de ferramentas tem sido desenvolvida para filtrá-los. Algumas das primeiras ferramentas desenvolvidas para eliminar os spam foram baseadas no Teorema de Bayes, tal como os **filtros de spam de Bayes.** Um filtro de spam bayesiano usa informações sobre e-mails previamente recebidos para saber se um e-mail recebido é um spam. Esses filtros procuram por ocorrências de determinadas palavras nas mensagens. Para determinada palavra w, a probabilidade de w aparecer em uma mensagem de spam é estimada pela determinação do número de vezes que w aparece em uma mensagem em um grande conjunto de mensagens conhecidas como spam e do número de vezes que ela aparece em um grande conjunto de mensagens conhecidas como não sendo spam. Quando examinamos as mensagens de e-mail para determinar se elas podem ser spam, procuramos por indicações de spam, tais como "oferta", "especial" ou "oportunidade", assim como as palavras que podem indicar que uma mensagem não é spam, tal como "mãe", "almoço" ou "Jan" (em que Jan é um de seus amigos). Infelizmente, os filtros de spam às vezes falham ao identificar uma mensagem de spam; isso é chamado de falso negativo. E, às vezes, identificam uma mensagem que não é spam, como spam; isto é chamado de falso positivo. Quando testada, é importante minimizar falsos positivos, porque filtrar os e-mails desejados é muito pior do que deixar algum spam passar.

Vamos desenvolver alguns filtros de spam bayesianos básicos. Primeiro, suponha que tenhamos um conjunto B de mensagens de spam e um conjunto G de mensagens que não são spam. (Por exemplo, os usuários podem classificar as mensagens como spam quando as examinam em suas caixas de entrada.) Depois, identificaremos as palavras que aparecem em B e em G. Contamos o números de mensagens no conjunto que contém cada palavra para encontrar $n_B(w)$ e $n_G(w)$, o número de mensagens com a palavra w nos conjuntos B e G, respectivamente. Então, a probabilidade empírica de um mensagem de spam conter a palavra w é $p(w) = n_B(w)/|B|$, e a probabilidade empírica de uma mensagem que não é spam conter a palavra w é $q(w) = n_G(w)/|G|$. Notamos que $p(w)$ e $q(w)$ estimam as probabilidades de uma mensagem recebida de spam e de uma mensagem recebida que não é spam conter a palavra w, respectivamente.

Agora suponha que recebamos um novo e-mail com a palavra w. Considere S como o evento de a mensagem ser spam. Considere E como o evento de a mensagem conter a palavra w. Os eventos S, que a mensagem é spam, e $\bar{S}$, que a mensagem não é spam, partem o conjunto de todas as mensagens. Assim, pelo Teorema de Bayes, a probabilidade de a mensagem ser spam, dado que ela tenha a palavra w, é

$$p(S \mid E) = \frac{p(E \mid S)p(S)}{p(E \mid S)p(S) + p(E \mid \bar{S})p(\bar{S})}.$$

Para aplicar essa fórmula, primeiro fazemos a estimativa de $p(S)$, a probabilidade de uma mensagem recebida ser spam, assim como de $p(\bar{S})$, a probabilidade de uma mensagem recebida não ser spam. Sem conhecimento anterior sobre a possibilidade de uma mensagem recebida ser spam, para simplificar, assumimos que a possibilidade de a mensagem ser ou não spam é a mesma. Ou seja, assumimos que $p(S) = p(\bar{S}) = 1/2$. Usando essa hipótese, encontramos que a probabilidade de uma mensagem ser spam, dado que ela tenha a palavra w, é

$$p(S \mid E) = \frac{p(E \mid S)}{p(E \mid S) + p(E \mid \bar{S})}.$$

(Note que, se temos algum dado empírico sobre a razão das mensagens de spam e das mensagens que não são spam, podemos mudar essa hipótese para produzir uma estimativa melhor para $p(S)$ e para $p(\bar{S})$; veja o Exercício 22.)

Depois, fazemos a estimativa de $p(E \mid S)$, a probabilidade condicional de a mensagem conter a palavra w, dado que a mensagem é spam, por $p(w)$. Da mesma forma, estimamos $p(E \mid \bar{S})$, a

probabilidade condicional de a mensagem conter a palavra w, dado que a mensagem não é spam, por $q(w)$. Inserindo essas estimativas para $p(E \mid S)$ e $p(E \mid \bar{S})$, temos que $p(S \mid E)$ pode ser estimado por

$$r(w) = \frac{p(w)}{p(w) + q(w)};$$

ou seja, $r(w)$ dá a estimativa da probabilidade de a mensagem ser spam, dado que ela contém a palavra w. Se $r(w)$ for maior que um percentual mínimo que executamos, tal como 0,9, então classificamos a mensagem como spam.

EXEMPLO 3 Suponha que encontremos que a palavra "Rolex" aparece em 250 das 2000 mensagens que são spam e em 5 das 1000 mensagens que não são spam. Faça a estimativa da probabilidade de uma mensagem recebida que contém a palavra "Rolex" ser spam, assumindo que é igualmente provável uma mensagem recebida ser ou não spam. Se nosso percentual mínimo para rejeitar a mensagem como spam é de 0,9, você rejeitará essa mensagem?

Solução: Usamos a contagem das quantidades de vezes que a palavra "Rolex" aparece em mensagens que são spam e nas que não são spam para encontrar $r(\text{Rolex}) = 250/2000 = 0{,}125$ e $q(\text{Rolex}) = 5/1000 = 0{,}005$. Como assumimos que é igualmente provável a mensagem recebida ser ou não spam, podemos estimar a probabilidade de a mensagem ser spam por

$$r(\text{Rolex}) = \frac{p(\text{Rolex})}{p(\text{Rolex}) + q(\text{Rolex})} = \frac{0{,}125}{0{,}125 + 0{,}005} = \frac{0{,}125}{0{,}130} \approx 0{,}962.$$

Como $r(\text{Rolex})$ é maior que o percentual mínimo de 0,9, rejeitamos esta mensagem como spam. ◀

Detectar um spam baseado na presença de uma única palavra pode nos levar a excessivos falsos positivos e falsos negativos. Conseqüentemente, os filtros de spam examinam a presença de várias palavras. Por exemplo, suponha que a mensagem contenha as palavras w_1 e w_2. Considere E_1 e E_2 como os eventos das mensagens que contêm as palavras w_1 e w_2, respectivamente. Para tornar nossas computações mais simples, assumimos que E_1 e E_2 são eventos independentes e que $E_1 \mid S$ e $E_2 \mid S$ são eventos independentes e que não temos conhecimento a priori se a mensagem é ou não spam. (As hipóteses de que E_1 e E_2 são independentes e que $E_1 \mid S$ e $E_2 \mid S$ são independentes podem introduzir alguns erros em nossas computações; assumimos que todos esses erros são pequenos.) Usando o Teorema de Bayes e nossas hipóteses, podemos mostrar que (veja o Exercício 23) $p(S \mid E_1 \cap E_2)$, a probabilidade de a mensagem ser spam, dado que ela contém as palavras w_1 e w_2, é

$$p(S \mid E_1 \cap E_2) = \frac{p(E_1 \mid S)p(E_2 \mid S)}{p(E_1 \mid S)p(E_2 \mid S) + p(E_1 \mid \bar{S})p(E_2 \mid \bar{S})}.$$

Estimamos a probabilidade $p(S \mid E_1 \cap E_2)$ por

$$r(w_1, w_2) = \frac{p(w_1)p(w_2)}{p(w_1)p(w_2) + q(w_1)q(w_2)}.$$

Ou seja, $r(w_1,w_2)$ estima a probabilidade de a mensagem ser spam, dado que ela contém as palavras w_1 e w_2. Quando $r(w_1,w_2)$ é maior que um percentual mínimo, tal como 0,9, determinamos que a mensagem é provavelmente um spam.

EXEMPLO 4 Suponha que executemos um filtro de spam bayesiano em um conjunto de 2000 mensagens de spam e 1000 mensagens que não são spam. A palavra "estoque" aparece em 400 mensagens de spam e em 60 mensagens que não são spam, e a palavra "subestimado" aparece em 200 mensagens de spam e em 25 mensagens que não são spam. Faça a estimativa da probabilidade de uma

mensagem recebida que contém as duas palavras, "estoque" e "subestimado", ser spam, assumindo que não temos conhecimento se ela é spam. Vamos rejeitar a mensagem como spam quando encontrarmos o percentual mínimo de 0,9?

Solução: Usando a contagem de cada uma dessas palavras nas mensagens que são spam ou naquelas que não são, obtemos as seguintes estimativas: $p(\text{estoque}) = 400/2000 = 0{,}2$, $q(\text{estoque}) = 60/1000 = 0{,}06$, $p(\text{subestimado}) = 200/2000 = 0{,}1$, e $q(\text{subestimado}) = 25/1000 = 0{,}025$. Usando essas probabilidades, podemos estimar a probabilidade de que a mensagem seja spam por

$$r(\text{estoque, subestimado}) = \frac{p(\text{estoque})p(\text{subestimado})}{p(\text{estoque})p(\text{subestimado}) + q(\text{estoque})q(\text{subestimado})} = \frac{(0{,}2)(0{,}1)}{(0{,}2)(0{,}1) + (0{,}06)(0{,}025)} \approx 0{,}930.$$

Como estabelecemos o percentual mínimo para rejeitar as mensagens em 0,9, as mensagens serão rejeitados pelo filtro. ◀

Quanto mais palavras nós usamos para estimar a probabilidade de uma mensagem recebida ser spam, maiores são nossas chances de determinar corretamente se ela é spam. Em geral, se E_i for o evento de a mensagem ter a palavra w_i, assumindo que o número de mensagens recebidas de spam é aproximadamente igual ao número de mensagens recebidas que não são spam e que os eventos $E_i \mid S$ são independentes, então pelo Teorema de Bayes a probabilidade de uma mensagem conter todas as palavras w_i, $i = 1, 2, \ldots, k$, é

$$p(S \mid \bigcap_{i=1}^{k} E_i) = \frac{\prod_{i=1}^{k} p(E_i \mid S)}{\prod_{i=1}^{k} p(E_i \mid S) + \prod_{i=1}^{k} p(E_i \mid \bar{S})}.$$

Podemos estimar a probabilidade por

$$r(w_1, w_2, \ldots, w_k) = \frac{\prod_{i=1}^{k} p(w_i)}{\prod_{i=1}^{k} p(w_i) + \prod_{i=1}^{k} q(w_i)}.$$

Para um filtro de spam mais eficiente, escolhemos as palavras para as quais a probabilidade de cada uma dessas palavras aparecer em spam é ou muito alta ou muito baixa. Quando computamos este valor para uma determinada mensagem, rejeitamos a mensagem como spam se $r(w_1, w_2, \ldots, w_k)$ exceder um percentual mínimo de 0,9.

Outra maneira de melhorar a performance de um filtro de spam bayesiano é olhar para as probabilidades de pares determinados de palavras aparecerem em spam e em mensagens que não são spam. Nós, então, tratamos as ocorrências dessas palavras como se aparecessem em um único bloco, em vez de aparecerem como duas palavras separadas. Por exemplo, o bloco de palavras "aumento da performance" é mais provável de indicar spam, enquanto "performance operacional" indica uma mensagem que não é spam. Da mesma forma, podemos saber a possibilidade de uma mensagem ser spam examinando a estrutura da mensagem para determinar onde as palavras aparecem nela. Além disso, os filtros de spam olham para a aparência de certos tipos de cadeias de caracteres em vez de olhar apenas para as palavras. Por exemplo, uma mensagem com um endereço válido de e-mail de um de seus amigos é menos provável de ser spam (se não enviado por um vírus) do que um com endereço de e-mail originário de um país conhecido pela sua grande quantidade de spams. Há uma batalha em andamento entre as pessoas que criam spam e aquelas que tentam filtrá-los. Isto leva a muitas técnicas novas de enganar os filtros de spam, incluindo a inserção nas mensagens de spam de longas cadeias de palavras que aparecem em mensagens que não são spam, assim como a inclusão de palavras dentro de figuras. As técnicas que discutimos aqui são apenas os primeiros passos na luta contra o spam.

Exercícios

1. Suponha que E e F sejam eventos de um campo amostral e $p(E) = 1/3$, $p(F) = 1/2$ e $p(E \mid F) = 2/5$. Encontre $p(F \mid E)$.

2. Suponha que E e F sejam eventos de um campo amostral e $p(E) = 2/3$, $p(F) = 3/4$ e $p(F \mid E) = 5/8$. Encontre $p(E \mid F)$.

3. Suponha que Frida escolha uma bola selecionando primeiro uma de duas caixas aleatoriamente e, então, escolhendo uma bola desta caixa, também aleatoriamente. A primeira caixa contém duas bolas brancas e três bolas azuis, e a segunda caixa contém quatro bolas brancas e uma bola azul. Qual é a probabilidade de Frida escolher uma bola da primeira caixa se ela pegou uma bola azul?

4. Suponha que Ana escolha uma bola selecionando primeiro uma de duas caixas aleatoriamente e, então, escolhendo uma bola desta caixa. A primeira caixa contém três bolas laranjas e quatro bolas pretas, e a segunda caixa contém cinco bolas laranjas e seis bolas pretas. Qual é a probabilidade de Ana escolher uma bola da segunda caixa se ela pegou uma bola laranja?

5. Suponha que 8% dos atletas de corrida de bicicleta usem esteróides, que um ciclista que usa esteróides tenha um teste positivo para esteróides em 96% das vezes e que um ciclista que não usa esteróides tenha um teste positivo para esteróides em 9% das vezes. Qual é a probabilidade de um ciclista, selecionado aleatoriamente, que tenha teste positivo para esteróides realmente usar esteróides?

6. Quando um teste para esteróides é aplicado em jogadores de futebol, 98% dos jogadores que tomam esteróides têm teste positivo e 12% dos jogadores que não tomam esteróides têm teste positivo. Suponha que 5% dos jogadores tomem esteróides. Qual é a probabilidade de um jogador que tenha teste positivo realmente tomar esteróides?

7. Suponha que um teste para usuários de ópio tenha uma taxa de 2% de falso positivo e uma taxa de 5% de falso negativo, ou seja, 2% das pessoas que não usam ópio têm teste positivo e 5% de pessoas que usam ópio têm teste negativo para a droga. Além disso, suponha que 1% das pessoas realmente usem ópio.

a) Encontre a probabilidade de alguém que tem teste negativo para ópio não usar a droga.

b) Encontre a probabilidade de alguém que tem teste positivo para ópio usar de fato a droga.

8. Suponha que uma pessoa em 10 000 tem uma doença genética rara. Há um excelente teste para a doença; 99,9% das pessoas com a doença têm teste positivo e apenas 0,02% das que não tem a doença tem teste positivo.

a) Qual é a probabilidade de alguém que tem teste positivo ter a doença genética?

b) Qual é a probabilidade de alguém que tem teste negativo não ter a doença genética?

9. Suponha que 8% dos pacientes em uma clínica estão infectados com HIV. Além disso, suponha que, quando um teste de sangue para HIV é feito, 98% dos pacientes infectados tem teste positivo e 3% dos pacientes não infectados com a doença tem teste positivo. Qual é a probabilidade de

a) um paciente com teste positivo para HIV estar infectado?

b) um paciente com teste positivo para HIV não estar infectado?

c) um paciente com teste negativo para HIV estar infectado?

d) um paciente com teste negativo para HIV não estar infectado?

10. Suponha que 4% dos pacientes em uma clínica estão infectados com gripe aviária. Além disso, suponha que, quando um teste de sangue para gripe aviária é feito, 97% dos pacientes infectados têm teste positivo e 2% dos pacientes não infectados com a doença têm teste positivo. Qual é a probabilidade de

a) um paciente com teste positivo para gripe aviária estar infectado?

b) um paciente com teste positivo para gripe aviária não estar infectado?

c) um paciente com teste negativo para gripe aviária estar infectado?

d) um paciente com teste negativo para gripe aviária não estar infectado?

11. Uma companhia de eletrônicos está planejando lançar um novo celular. A companhia encarrega um agente de marketing para, a cada novo produto, prever se este será um sucesso ou um fracasso. Dos novos produtos lançados pela companhia, 60% tiveram sucesso. Além disso, 70% de seus produtos de sucesso produzidos foram avaliados para ter sucesso, enquanto 40% dos produtos que fracassaram foram avaliados para ter sucesso. Encontre a probabilidade de este novo celular ter sucesso se ele foi avaliado para ter sucesso.

***12.** Um satélite espacial próximo de Netuno comunica-se com a Terra usando cadeias de bits. Suponha que nessa transmissão, ele envie um 1 em um terço de tempo e um 0 em dois terços de tempo. Quando um 0 é enviado, a probabilidade de ele ser recebido corretamente é de 0,9, e a probabilidade de ele ser recebido incorretamente (como um 1) é de 0,1. Quando um 1 é enviado, a probabilidade de ser recebido corretamente é de 0,8, e a probabilidade de ele ser recebido incorretamente (como um 0) é de 0,2.

a) Encontre a probabilidade de um 0 ser recebido.

b) Use o Teorema de Bayes para encontrar a probabilidade de ser transmitido um 0, dado que um 0 foi recebido.

13. Suponha que E, F_1, F_2 e F_3 sejam eventos de um espaço amostral S e que F_1, F_2 e F_3 sejam mutuamente disjuntos e sua união seja S. Encontre $p(F_1 \mid E)$ se $p(E \mid F_1) = 1/8$, $p(E \mid F_2) = 1/4$, $p(E \mid F_3) = 1/6$, $p(F_1) = 1/4$, $p(F_2) = 1/4$ e $p(F_3) = 1/2$.

14. Suponha que E, F_1, F_2 e F_3 sejam eventos de um espaço amostral S e que F_1, F_2 e F_3 sejam mutuamente disjuntos e sua união seja S. Encontre $p(F_2 \mid E)$ se $p(E \mid F_1) = 2/7$,

$p(E \mid F_2) = 3/8$, $p(E \mid F_3) = 1/2$, $p(F_1) = 1/6$, $p(F_2) = 1/2$ e $p(F_3) = 1/3$.

15. Neste exercício, vamos usar o Teorema de Bayes para resolver o Problema de Monty Hall (Exemplo 10 da Seção 6.1). Relembre que, nesse problema, você tem de escolher uma das três portas para ser aberta. Há um grande prêmio atrás de uma delas e as outras duas não têm nada. Depois que você escolheu uma porta, Monty Hall abre uma das duas portas restantes que ele sabe que não tem nada, selecionando-a aleatoriamente se ambas não têm nada. Monty pergunta a você se não gostaria de trocar de porta. Suponha que as três portas no problema estejam numeradas como 1, 2 e 3. Considere W como a variável aleatória cujo valor é o número de portas com o prêmio; assuma que $p(W = k) = 1/3$ para $k = 1, 2, 3$. Considere M como a variável aleatória cujo valor é o número de portas que Monty abre. Suponha que você tenha escolhido a porta i.

a) Qual é a probabilidade de você ganhar o prêmio se o jogo termina depois que Monty pergunta se você quer trocar de porta?

b) Encontre $p(M = j \mid W = k)$ para $j = 1, 2, 3$ e $k = 1, 2, 3$.

c) Use o Teorema de Bayes para encontrar $p(W = j \mid M = k)$, em que i, j e k são valores distintos.

d) Explique por que a resposta do item (c) nos diz que você deveria ter trocado de porta quando Monty lhe deu a chance.

16. Ramesh pode ir trabalhar de três maneiras diferentes: de bicicleta, de carro ou de ônibus. Com o trânsito, há uma chance de 50% de ele chegar atrasado quando ele vai de carro. Quando ele pega o ônibus, que usa uma faixa especial reservada para ônibus, há uma chance de 20% de chegar atrasado. A probabilidade de ele chegar atrasado quando vai de bicicleta é apenas de 5%. Ramesh chega atrasado apenas um dia. Seu chefe quer fazer a estimativa da probabilidade de ele ter ido de carro aquele dia.

a) Suponha que o chefe assuma que há uma chance de 1/3 de Ramesh escolher cada uma das três maneiras de ir para o trabalho. Que estimativa para a probabilidade de Ramesh ter ido de carro o chefe obtém a partir do Teorema de Bayes, considerando sua hipótese?

b) Suponha que o chefe saiba que Ramesh vai de carro 30% das vezes, pega o ônibus apenas 10% das vezes e vai de bicicleta 60% das vezes. Qual estimativa para a probabilidade de Ramesh ter ido de carro o chefe obtém a partir do Teorema de Bayes, usando essa informação?

***17.** Prove o Teorema 2, a extensão do Teorema de Bayes, ou seja, suponha que E seja um evento do espaço amostral S e que $F_1, F_2, \ldots, F_n$ sejam eventos mútuos exclusivos, tal que $\bigcup_{i=1}^{n} F_i = S$. Assuma que $p(E) \neq 0$ e $p(F_i) \neq 0$ para $i = 1, 2, \ldots, n$. Mostre que

$$p(F_j \mid E) = \frac{p(E \mid F_j)p(F_j)}{\sum_{i=1}^{n} p(E \mid F_i)p(F_i)}.$$

[*Dica:* Use o fato de que $E = \bigcup_{i=1}^{n} (E \cap F_i)$.]

18. Suponha que um filtro de spam bayesiano é executado em um conjunto de 500 mensagens de spam e 200 mensagens que não são spam. A palavra "excitante" aparece em 40 mensagens de spam e em 25 mensagens que não são spam. Uma mensagem recebida pode ser rejeitada como spam se ela tiver a palavra "excitante" e o percentual mínimo para rejeitar um spam for 0,9?

19. Suponha que um filtro de spam bayesiano é executado em um conjunto de 1000 mensagens de spam e 400 mensagens que não são spam. A palavra "oportunidade" aparece em 175 mensagens de spam e em 20 mensagens que não são spam. Uma mensagem recebida pode ser rejeitada como spam se ela tiver a palavra "oportunidade" e o percentual mínimo para rejeitar um spam for 0,9?

20. Podemos rejeitar a mensagem como spam do Exemplo 4

a) usando apenas o fato de que a palavra "subestimado" ocorre na mensagem?

b) usando apenas o fato de que a palavra "estoque" ocorre na mensagem?

21. Suponha que um filtro de spam bayesiano é executado em um conjunto de 10 000 mensagens de spam e 5000 mensagens que não são spam. A palavra "aprimoramento" apareça em 1500 mensagens de spam e em 20 mensagens que não são spam, enquanto a palavra "herbal" aparece em 800 mensagens de spam e em 200 mensagens que não são spam. Faça a estimativa da probabilidade de uma mensagem recebida, que contenha as palavras "aprimoramento" e "herbal", ser spam. A mensagem será rejeitada como spam se o percentual mínimo para rejeitar um spam for 0,9?

22. Suponha que tenhamos a informação de que uma mensagem qualquer recebida é spam. Em particular, suponha que durante um período de tempo encontramos que s mensagens de spam chegam e h mensagens que não são spam chegam também.

a) Use essa informação para estimar $p(S)$, a probabilidade de uma mensagem recebida ser spam, e $p(\bar{S})$, a probabilidade de uma mensagem recebida não ser spam.

b) Use o Teorema de Bayes e o item (a) para estimar a probabilidade de uma mensagem recebida que contém a palavra w ser spam, em que $p(w)$ é a probabilidade de w ocorrer em uma mensagem de spam e $q(w)$ é a probabilidade de w ocorrer em uma mensagem que não é spam.

23. Suponha que E_1 e E_2 sejam os eventos de uma mensagem recebida conter as palavras w_1 e w_2, respectivamente. Assumindo que E_1 e E_2 são eventos independentes e que $E_1 \mid S$ e $E_2 \mid S$ são eventos independentes, em que S é o evento de uma mensagem recebida ser spam, e que não temos conhecimento se a mensagem é ou não spam, mostre que

$$p(S \mid E_1 \cap E_2) = \frac{p(E_1 \mid S)p(E_2 \mid S)}{p(E_1 \mid S)p(E_2 \mid S) + p(E_1 \mid \bar{S})p(E_2 \mid \bar{S})}.$$

6.4 Valor Esperado e Variância

Introdução

O **valor esperado** de uma variável aleatória é a soma sobre todos os elementos em um espaço amostral do produto da probabilidade do elemento e do valor da variável aleatória relacionada a esse elemento. Conseqüentemente, o valor esperado é uma média ponderada dos valores de uma variável aleatória. O valor esperado de uma variável aleatória fornece um ponto central para a distribuição de valores dessa variável aleatória. Podemos resolver muitos problemas usando a noção de valor esperado de uma variável aleatória, tal como determinar se temos vantagem em apostas ou computar a complexidade de caso médio de algoritmos. Outra medida útil de uma variável aleatória é sua **variância,** que nos diz como estão organizados os valores dessa variável aleatória. Podemos usar a variância de uma variável aleatória para nos ajudar a estimar a probabilidade de uma variável aleatória adquirir valores afastados de seu valor esperado.

Valores Esperados

Links

Muitas questões podem ser formuladas em termos do valor que esperamos que uma variável aleatória pode adquirir, ou mais precisamente, o valor médio de uma variável aleatória quando um experimento é realizado um grande número de vezes. Questões deste tipo incluem: Quantas caras esperamos que apareçam quando uma moeda é lançada 100 vezes? Qual é a expectativa do número de comparações usadas para encontrar um elemento em uma lista usando a busca linear? Para estudar tais questões, introduziremos o conceito de valor esperado de uma variável aleatória.

DEFINIÇÃO 1

O *valor esperado* (ou *esperança*) de uma variável aleatória $X(s)$ em um espaço amostral S é igual a

$$E(X) = \sum_{s \in S} p(s)X(s).$$

Note que quando o espaço amostral S tem n elementos $S = \{x_1, x_2, \ldots, x_n\}$, $E(X) = \sum_{i=1}^{n} p(x_i)X(x_i)$.

Lembre-se: Quando há um número muito grande de elementos no espaço amostral, a esperança é definida apenas quando a série infinita da definição é absolutamente convergente. Em particular, a esperança de uma variável aleatória no espaço amostral é finita, se ela existir.

EXEMPLO 1 **Valor Esperado de um Dado** Considere X como o número de resultados quando um dado é lançado. Qual é o valor esperado de X?

Solução: A variável aleatória X adquire os valores 1, 2, 3, 4, 5 ou 6, cada um com probabilidade $1/6$. Temos que

$$E(X) = \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{6} \cdot 5 + \frac{1}{6} \cdot 6 = \frac{21}{6} = \frac{7}{2}.$$

◀

EXEMPLO 2 Uma moeda não-viciada é lançada três vezes. Considere S como o espaço amostral dos oito possíveis resultados e considere X como a variável aleatória que designa a um resultado o número de caras desse resultado. Qual é o valor esperado de X?

Exemplos Extras

Solução: No Exemplo 10 da Seção 6.2, listamos os valores de X para os oito possíveis resultados quando uma moeda é lançada três vezes. Como a moeda não é viciada e as jogadas são independentes, a probabilidade de cada resultado é $1/8$. Conseqüentemente,

$$\begin{aligned} E(X) &= \frac{1}{8}[X(HHH) + X(HHT) + X(HTH) + X(THH) + X(TTH) \\ &\quad + X(THT) + X(HTT) + X(TTT)] \\ &= \frac{1}{8}(3+2+2+2+1+1+1+0) = \frac{12}{8} \\ &= \frac{3}{2}. \end{aligned}$$

Conseqüentemente, o número esperado de caras em três lançamentos de uma moeda não-viciada é de $3/2$. ◀

Quando um experimento tem relativamente poucos resultados, podemos computar o valor esperado de uma variável aleatória diretamente a partir de sua definição, como foi feito no Exemplo 2. Entretanto, quando um experimento tem um grande número de resultados, pode ser inconveniente computar o valor esperado de uma variável aleatória a partir de sua definição. Em vez disso, podemos encontrar o valor esperado de uma variável aleatória agrupando todos os resultados determinados para o mesmo valor pela variável aleatória, como mostra o Teorema 1.

TEOREMA 1 Se X é uma variável aleatória e $p(X = r)$ é a probabilidade de $X = r$, para que $p(X = r) = \sum_{s \in S,\, X(s) = r} p(s)$, então

$$E(X) = \sum_{r \in X(S)} p(X = r)\, r.$$

Demonstração: Suponha que X seja uma variável aleatória com média $X(S)$ e considere $p(X = r)$ como a probabilidade de a variável aleatória X admitir o valor r. Conseqüentemente, $p(X = r)$ é a soma das probabilidades dos resultados s, tal que $X(s) = r$. Temos que

$$E(X) = \sum_{r \in X(S)} p(X = r) r.$$ ◁

O Exemplo 3 e a demonstração do Teorema 2 ilustram o uso dessa fórmula. No Exemplo 3, encontraremos o valor esperado da soma dos números que aparecem em dois dados não-viciados quando eles são jogados. No Teorema 2, vamos encontrar o valor esperado do número de sucessos quando n testes de Bernoulli são realizados.

EXEMPLO 3 Qual é o valor esperado da soma dos números que aparecem quando um par de dados não-viciados é lançado?

Solução: Considere X como a variável aleatória que é igual à soma dos números que aparecem quando um par de dados é lançado. No Exemplo 12 da Seção 6.2, listamos o valor de X para os

36 resultados desse experimento. O domínio de X é $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$. Pelo Exemplo 12 da Seção 6.2, vemos que

$$p(X = 2) = p(X = 12) = 1/36,$$
$$p(X = 3) = p(X = 11) = 2/36 = 1/18,$$
$$p(X = 4) = p(X = 10) = 3/36 = 1/12,$$
$$p(X = 5) = p(X = 9) = 4/36 = 1/9,$$
$$p(X = 6) = p(X = 8) = 5/36,$$
$$p(X = 7) = 6/36 = 1/6.$$

Substituindo esses valores na fórmula, temos

$$\begin{aligned} E(X) &= 2 \cdot \frac{1}{36} + 3 \cdot \frac{1}{18} + 4 \cdot \frac{1}{12} + 5 \cdot \frac{1}{9} + 6 \cdot \frac{5}{36} + 7 \cdot \frac{1}{6} \\ &\quad + 8 \cdot \frac{5}{36} + 9 \cdot \frac{1}{9} + 10 \cdot \frac{1}{12} + 11 \cdot \frac{1}{18} + 12 \cdot \frac{1}{36} \\ &= 7. \end{aligned}$$

◀

TEOREMA 2 O número esperado de sucessos quando n testes independentes de Bernoulli são realizados, em que p é a probabilidade de sucesso em cada teste, é np.

Demonstração: Considere X como a variável aleatória igual ao número de sucessos em n testes. Pelo Teorema 2 da Seção 6.2, vemos que $p(X = k) = C(n, k)p^k q^{n-k}$. Assim, temos

$$\begin{aligned} E(X) &= \sum_{k=1}^{n} kp(X = k) && \text{pelo Teorema 1} \\ &= \sum_{k=1}^{n} kC(n,k)p^k q^{n-k} && \text{pelo Teorema 2 da Seção 6.2} \\ &= \sum_{k=1}^{n} nC(n-1, k-1)p^k q^{n-k} && \text{pelo Exercício 21 da Seção 5.4} \\ &= np\sum_{k=1}^{n} C(n-1, k-1)p^{k-1} q^{n-k} && \text{fatorando } np \text{ a partir de cada termo} \\ &= np\sum_{j=0}^{n-1} C(n-1, j)p^j q^{n-1-j} && \text{substituindo o índice da somatória com } j = k-1 \\ &= np(p+q)^{n-1} && \text{pelo Teorema Binomial} \\ &= np. && \text{porque } p + q = 1 \end{aligned}$$

Isso completa a demonstração, pois mostra que o número esperado de sucesso em n testes independentes de Bernoulli é np. ◁

Agora vamos mostrar que a hipótese de os testes de Bernoulli serem independentes no Teorema 2 não é necessária.

Linearidade da Esperança

O Teorema 3 nos diz que os valores esperados são lineares. Por exemplo, o valor esperado da soma de variáveis aleatórias é a soma de seus valores esperados. Acharemos esta propriedade extremamente útil.

TEOREMA 3 Se X_i, $i = 1, 2, \ldots, n$ com n como um número inteiro positivo, forem variáveis aleatórias em S, e se a e b forem números reais, então

(*i*) $E(X_1 + X_2 + \cdots + X_n) = E(X_1) + E(X_2) + \cdots + E(X_n)$

(*ii*) $E(aX + b) = aE(X) + b$.

Demonstração: O primeiro resultado para $n = 2$ é obtido diretamente a partir da definição de valor esperado, porque

$$\begin{aligned} E(X_1 + X_2) &= \sum_{s \in S} p(s)(X_1(s) + X_2(s)) \\ &= \sum_{s \in S} p(s)X_1(s) + \sum_{s \in S} p(s)X_2(s) \\ &= E(X_1) + E(X_2). \end{aligned}$$

O caso com n variáveis aleatórias é obtido facilmente usando a indução matemática a partir do caso de duas variáveis aleatórias. Por fim, note que

$$\begin{aligned} E(aX + b) &= \textstyle\sum_{s \in S} p(s)(aX(s) + b) \\ &= a \textstyle\sum_{s \in S} p(s)X(s) + b \sum_{s \in S} p(s) \\ &= a\,E(X) + b \text{ porque } \textstyle\sum_{s \in S} p(s) = 1. \end{aligned}$$

◁

Os exemplos 4 e 5 ilustram como usar o Teorema 3.

EXEMPLO 4 Use o Teorema 3 para encontrar o valor esperado da soma dos números que aparecem quando um par de dados não-viciados é jogado. (Isto foi feito no Exemplo 3, sem o benefício deste teorema.)

Solução: Considere X_1 e X_2 como variáveis aleatórias com $X_1((i, j)) = i$ e $X_2((i, j)) = j$, para que X_1 seja o número que aparece no primeiro dado e X_2, o número que aparece no segundo dado. É fácil ver que $E(X_1) = E(X_2) = 7/2$ porque ambos são iguais $(1 + 2 + 3 + 4 + 5 + 6)/6 = 21/6 = 7/2$. A soma dos dois números que aparecem quando os dois dados são jogados é a soma $X_1 + X_2$. Pelo Teorema 3, o valor esperado da soma é $E(X_1 + X_2) = E(X_1) + E(X_2) = 7/2 + 7/2 = 7$. ◀

EXEMPLO 5 Na demonstração do Teorema 2, encontramos o valor esperado do número de sucessos quando n testes independentes de Bernoulli são realizados, em que p é a probabilidade de sucesso em cada

teste pela computação direta. Mostre como o Teorema 3 pode ser usado para derivar este resultado onde os testes de Bernoulli não são necessariamente independentes.

Solução: Considere X_i como a variável aleatória com $X_i((t_1, t_2, \ldots, t_n)) = 1$ se t_i for um sucesso, e $X_i((t_1, t_2, \ldots, t_n)) = 0$ se t_i for um fracasso. O valor esperado de X_i é $E(X_i) = 1 \cdot p + 0 \cdot (1-p) = p$ para $i = 1, 2, \ldots, n$. Considere $X = X_1 + X_2 + \cdots + X_n$, para que X faça a contagem de número de sucessos, quando estes n testes de Bernoulli são realizados. O Teorema 3, aplicado à soma de n variáveis aleatórias, mostra que $E(X) = E(X_1) + E(X_2) + \cdots + E(X_n) = np$. ◀

Podemos tirar vantagem da linearidade das esperanças para encontrar as soluções de muitos problemas difíceis semelhantes. O principal passo é expressar uma variável aleatória cuja esperança desejamos encontrar como a soma das variáveis aleatórias, de modo que as esperanças dessas variáveis sejam mais fáceis de encontrar. Os exemplos 6 e 7 ilustram esta técnica.

EXEMPLO 6 **Valor Esperado no Problema do Chapéu** Um novo empregado recebe os chapéus de n pessoas em um restaurante, esquecendo de numerá-los. Quando os clientes retornam para pegar seus chapéus, o guardador dá a eles os chapéus que escolhe aleatoriamente a partir dos restantes. Qual é o número esperado de chapéus que são devolvidos corretamente?

Solução: Considere X como a variável aleatória que é igual ao número de pessoas que receberam o chapéu corretamente. Considere X_i como a variável aleatória com $X_i = 1$ se a i-ésima pessoa receber o chapéu correto e $X_i = 0$ se ocorrer o contrário. Temos que

$$X = X_1 + X_2 + \cdots + X_n.$$

Como é igualmente provável que o guardador devolva qualquer um dos chapéus a essa pessoa, temos que a probabilidade de a i-ésima pessoa receber o chapéu correto é de $1/n$. Conseqüentemente, pelo Teorema 1, para todo i, temos que

$$E(X_i) = 1 \cdot p(X_i = 1) + 0 \cdot p(X_i = 0) = 1 \cdot 1/n + 0 = 1/n.$$

Pela linearidade da esperança (Teorema 3), temos que

$$E(X) = E(X_1) + E(X_2) + \cdots + E(X_n) = n \cdot 1/n = 1.$$

Conseqüentemente, a média do número de pessoas que recebem o chapéu correto é exatamente 1. Note que esta resposta é independente do número de pessoas que tiveram seus chapéus guardados! (Encontraremos uma fórmula explícita para a probabilidade de ninguém receber o chapéu correto na Seção 7.6.) ◀

EXEMPLO 7 **Número Esperado de Inversões em uma Permutação** O par ordenado (i, j) é chamado de **inverso** na permutação dos n primeiros números inteiros positivos se $i < j$, mas j precede i na permutação. Por exemplo, há seis inversões na permutação 3, 5, 1, 4, 2; essas inversões são

$$(1, 3), (1, 5), (2, 3), (2, 4), (2, 5), (4, 5).$$

Para encontrar o número esperado de inversões em uma permutação aleatória dos n primeiros números inteiros positivos, tomamos $I_{i,j}$ como a variável aleatória do conjunto de todas as permutações dos primeiros n números inteiros positivos com $I_{i,j} = 1$ se (i, j) for uma inversão da permutação e $I_{i,j} = 0$ se ocorrer o contrário. Temos que se X for uma variável aleatória igual ao número de inversões na permutação, então

$$X = \sum_{1 \le i < j \le n} I_{i,j}.$$

Note que é igualmente provável para i preceder j em uma permutação escolhida aleatoriamente como é igualmente provável para j preceder i. (Para ver isso, note que há um número igual de permutações com cada uma dessas propriedades.) Conseqüentemente, para os pares de i e j, temos

$$E(I_{i,j}) = 1 \cdot p(I_{i,j} = 1) + 0 \cdot p(I_{i,j} = 0) = 1 \cdot 1/2 + 0 = 1/2.$$

Como há $\binom{n}{2}$ pares de i e j com $1 \leq i < j \leq n$ e pela linearidade da esperança (Teorema 3), temos

$$E(X) = \sum_{1 \leq i < j \leq n} I_{i,j} = \binom{n}{2} \cdot \frac{1}{2} = \frac{n(n-1)}{4}.$$

Temos que há uma média de $n(n-1)/4$ inversões em uma permutação dos primeiros n números inteiros positivos. ◀

Complexidade Computacional de Caso Médio

Links Computar a complexidade computacional de caso médio de um algoritmo pode ser interpretado como computar o valor esperado de uma variável aleatória. Considere o espaço amostral de um experimento como o conjunto de possíveis entradas a_j, $j = 1, 2, \ldots, n$, e considere X como a variável aleatória que designa para a_j o número de operações usadas pelo algoritmo quando a_j é dado como uma entrada. Baseados em nosso conhecimento sobre entradas, determinamos a probabilidade $p(a_j)$ de cada possível valor de entrada a_j. Então, a complexidade de caso médio do algoritmo é

$$E(X) = \sum_{j=1}^{n} p(a_j) X(a_j).$$

Este é o valor esperado de X.

Encontrar a complexidade computacional de caso médio de um algoritmo geralmente é muito mais difícil do que encontrar a complexidade computacional de caso pior e, na maioria das vezes, envolve o uso de métodos sofisticados. Entretanto, há alguns algoritmos para os quais a análise necessária para encontrar a complexidade computacional de caso médio não é difícil. No Exemplo 8 ilustraremos como encontrar a complexidade computacional de caso médio do algoritmo de busca linear considerando diferentes hipóteses para a probabilidade de o elemento, pelo qual buscamos, ser um elemento da lista.

EXEMPLO 8 **Complexidade de Caso Médio do Algoritmo de Busca Linear** Fornecemos um número real x e uma lista de n números reais distintos. O algoritmo de busca linear, descrito na Seção 3.1, localiza x por comparações sucessivas deste com cada um dos elementos na lista, finalizando quando x for localizado ou quando todos os elementos forem examinados e se verificar que x não está na lista. Qual é a complexidade computacional de caso médio do algoritmo de busca linear se a probabilidade de x estar na lista é p e é igualmente provável que x seja qualquer um dos n elementos da lista? (Há $n + 1$ tipos possíveis de entrada: os n números na lista e um número que não está na lista, que tratamos como uma entrada única.)

Solução: No Exemplo 4 da Seção 3.3, mostramos que $2i + 1$ comparações são usadas se x for igual ao i-ésimo elemento da lista e, no Exemplo 2 da Seção 3.3, que $2n + 2$ comparações são usadas se x não estiver na lista. A probabilidade de x ser igual a a_i, o i-ésimo elemento da lista, é

p/n, e a probabilidade de x não estar na lista é $q = 1 - p$. Temos que a complexidade computacional de caso médio do algoritmo de busca linear é

$$\begin{aligned} E &= \frac{3p}{n} + \frac{5p}{n} + \cdots + \frac{(2n+1)p}{n} + (2n+2)q \\ &= \frac{p}{n}(3 + 5 + \cdots + (2n+1)) + (2n+2)q \\ &= \frac{p}{n}((n+1)^2 - 1) + (2n+2)q \\ &= p(n+2) + (2n+2)q. \end{aligned}$$

(A terceira igualdade foi obtida a partir do Exemplo 2 da Seção 4.1.) Por exemplo, quando é garantido que x está na lista, temos $p = 1$ (então, a probabilidade de $x = a_i$ é $1/n$ para cada i) e $q = 0$. Então, $E = n + 2$, como mostramos no Exemplo 4 da Seção 3.3.

Quando p, a probabilidade de x estar na lista, é $1/2$, temos que $q = 1 - p = 1/2$, assim, $E = (n+2)/2 + n + 1 = (3n+4)/2$. Da mesma forma, se a probabilidade de x estar na lista é $3/4$, temos $p = 3/4$ e $q = 1/4$, assim, $E = 3(n+2)/4 + (n+1)/2 = (5n+8)/4$.

Por fim, quando é garantido que x não esteja na lista, temos $p = 0$ e $q = 1$. Temos que $E = 2n + 2$, o que não é surpreendente porque temos de procurar por toda a lista. ◀

O Exemplo 9 ilustra como a linearidade da esperança pode nos ajudar a encontrar a complexidade de caso médio de um algoritmo de ordenação, o insertion sort.

EXEMPLO 9 **Complexidade de Caso Médio do Insertion Sort** Qual é a média do número de comparações usadas pelo insertion sort para organizar n elementos distintos?

Solução: Primeiro, suponha que X seja a variável aleatória igual ao número de comparações usadas pelo insertion sort (descrito na Seção 3.1) para organizar a lista $a_1, a_2, \ldots, a_n$ de n elementos distintos. Então, $E(X)$ é o número médio de comparações usadas. (Lembre que no passo i para $i = 2, \ldots, n$, o insertion sort insere o i-ésimo elemento da lista original na posição correta na lista ordenada dos primeiros $i - 1$ elementos da lista original.)

Consideremos X_i como a variável aleatória igual ao número de comparações usadas para inserir a_i na devida posição depois que os primeiros $i - 1$ elementos $a_1, a_2, \ldots, a_{i-1}$ foram organizados. Como

$$X = X_2 + X_3 + \cdots + X_n,$$

podemos usar a linearidade da esperança para concluir que

$$E(X) = E(X_2 + X_3 + \cdots + X_n) = E(X_2) + E(X_3) + \cdots + E(X_n).$$

Para encontrar $E(X_i)$ para $i = 2, 3, \ldots, n$, considere $p_j(k)$ como a probabilidade de o maior dos primeiros j elementos da lista aparecer na k-ésima posição, ou seja, de $\max(a_1, a_2, \ldots, a_j) = a_k$, em que $1 \leq k \leq j$. Como os elementos da lista são distribuídos aleatoriamente, é igualmente provável para o maior elemento entre os primeiros j elementos aparecer em qualquer posição. Conseqüentemente, $p_j(k) = 1/j$. Se $X_i(k)$ for igual ao número de comparações usadas pelo insertion sort se a_i for inserido na k-ésima posição da lista uma vez que $a_1, a_2, \ldots, a_{i-1}$ já tiverem sido

ordenados, temos que $X_i(k) = k$. Como é possível que a_i seja inserido em qualquer das primeiras i posições, encontramos que

$$E(X_i) = \sum_{k=1}^{i} p_i(k) \cdot X_i(k) = \sum_{k=1}^{i} \frac{1}{i} \cdot k = \frac{1}{i} \cdot \sum_{k=1}^{i} k = \frac{1}{i} \cdot \frac{i(i+1)}{2} = \frac{i+1}{2}.$$

Daí segue que

$$\begin{aligned} E(X) &= \sum_{i=2}^{n} E(X_i) = \sum_{i=2}^{n} \frac{i+1}{2} = \frac{1}{2} \sum_{j=3}^{n+1} j \\ &= \frac{1}{2} \frac{(n+1)(n+2)}{2} - \frac{1}{2}(1+2) = \frac{n^2+3n-4}{4}. \end{aligned}$$

Para obter a terceira dessas igualdades, substituímos o índice da somatória, fazendo $j = i + 1$. Para obter a quarta igualdade, usamos a fórmula $\sum_{k=1}^{m} k = m(m+1)/2$ (da Tabela 2 da Seção 2.4) com $m = n + 1$, subtraindo os termos que não aparecem com $j = 1$ e $j = 2$. Concluímos que o número médio de comparações usado pelo insertion sort para organizar n elementos é igual a $(n^2 + 3n - 4)/4$, que é $\Theta(n^2)$. ◀

A Distribuição Geométrica

Mudaremos agora nosso foco para uma variável aleatória com número infinito de resultados.

EXEMPLO 10 Suponha que a probabilidade de uma moeda virar coroa é p. Essa moeda é lançada repetidas vezes até que saia uma coroa. Qual é o número esperado de jogadas para que saia a primeira coroa na moeda?

Links *Solução:* Primeiro notamos que o espaço amostral consiste em todas as seqüências que começam com qualquer número de caras, indicadas por H, seguidas de uma coroa, indicada por T. Assim, o espaço amostral é o conjunto $\{T, HT, HHT, HHHT, HHHHT, \ldots\}$. Note que esse é um espaço amostral infinito. Podemos determinar a probabilidade de um elemento do espaço amostral notando que os lançamentos da moeda são independentes e que a probabilidade de sair uma cara é $1 - p$. Assim, $p(T) = p$, $p(HT) = (1-p)p$, $p(HHT) = (1-p)^2 p$, e, em geral, a probabilidade de uma moeda ser lançada n vezes antes de sair uma coroa, ou seja, de sair $n-1$ caras seguida de uma coroa, é $(1-p)^{n-1}p$. (O Exercício 14 pede para mostrar que a soma das probabilidades dos pontos no espaço amostral é 1.)

Agora considere X como a variável aleatória igual ao número de lançamentos de um elemento no espaço amostral. Ou seja, $X(T) = 1$, $X(HT) = 2$, $X(HHT) = 3$, e assim por diante. Note que $p(X = j) = (1-p)^{j-1}p$. O número esperado de jogadas até que saia uma coroa na moeda é igual a $E(X)$.

Usando o Teorema 1, encontramos que

$$E(X) = \sum_{j=1}^{\infty} j \cdot p(X = j) = \sum_{j=1}^{\infty} j(1-p)^{j-1}p = p\sum_{j=1}^{\infty} j(1-p)^{j-1} = p \cdot \frac{1}{p^2} = \frac{1}{p}.$$

[A terceira igualdade nesta cadeia é obtida a partir da Tabela 2 da Seção 2.4, que nos diz que $\sum_{j=1}^{\infty} j(1-p)^{j-1} = 1/(1-(1-p))^2 = 1/p^2$.] Temos que o número esperado de vezes que uma moeda é jogada até que saia uma coroa é $1/p$. Note que quando uma moeda não é viciada temos $p = 1/2$, assim, o número esperado de lançamentos até que saia uma coroa é $1/(1/2) = 2$.

◀

A variável aleatória X que se iguala ao número de jogadas esperadas até que saia uma coroa é um exemplo de uma variável aleatória com uma **distribuição geométrica.**

DEFINIÇÃO 2 Uma variável aleatória X tem uma *distribuição geométrica de parâmetro* p se $p(X = k) = (1 - p)^{k-1} p$ para $k = 1, 2, 3, \ldots$.

As distribuições geométricas aparecem em muitas aplicações porque elas são usadas para estudar o tempo necessário antes que um determinado evento aconteça, tal como o tempo necessário antes de encontrarmos um objeto com determinada propriedade, o número de tentativas antes de um experimento ser bem-sucedido, o número de vezes que um produto pode ser usado antes de falhar, e assim por diante.

Quando computamos o valor esperado do número de lançamentos necessários antes de sair coroa, demonstramos o Teorema 4.

TEOREMA 4 Se a variável aleatória X tem distribuição geométrica de parâmetro p, então $E(X) = 1/p$.

Variáveis Aleatórias Independentes

Já discutimos sobre eventos independentes. Vamos agora definir o que significa duas variáveis aleatórias serem independentes.

DEFINIÇÃO 3 As variáveis aleatórias X e Y de um espaço amostral S são *independentes* se

$$p(X(s) = r_1 \text{ e } Y(s) = r_2) = p(X(s) = r_1) \cdot p(Y(s) = r_2),$$

ou discursivamente, se a probabilidade de $X(s) = r_1$ e $Y(s) = r_2$ for igual ao produto das probabilidades de $X(s) = r_1$ e $Y(s) = r_2$, para todos os números reais r_1 e r_2.

EXEMPLO 11 As variáveis aleatórias X_1 e X_2 do Exemplo 4 são independentes?

Exemplos Extras

Solução: Considere $S = \{1, 2, 3, 4, 5, 6\}$, e assuma $i \in S$ e $j \in S$. Como há 36 resultados possíveis quando um par de dados é lançado e cada um é igualmente possível de aparecer, temos

$$p(X_1 = i \text{ e } X_2 = j) = 1/36.$$

Além disso, $p(X_1 = i) = 1/6$ e $p(X_2 = j) = 1/6$, pois a probabilidade de i aparecer no primeiro dado e a probabilidade de j aparecer no segundo dado são ambas $1/6$. Temos que

$$p(X_1 = i \text{ e } X_2 = j) = \frac{1}{36} \quad \text{e} \quad p(X_i = i)p(X_2 = j) = \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36},$$

assim, X_1 e X_2 são independentes. ◀

EXEMPLO 12 Mostre que as variáveis aleatórias X_1 e $X = X_1 + X_2$, em que X_1 e X_2 são definidos como no Exemplo 4, não são independentes.

Solução: Note que $p(X_1 = 1 \text{ e } X = 12) = 0$, pois $X_1 = 1$ significa que o número que aparece no primeiro dado é 1, implicando que a soma dos números que aparecem nos dois dados não pode ser igual a 12. Por outro lado, $p(X_1 = 1) = 1/6$ e $p(X = 12) = 1/36$. Assim, $p(X_1 = 1 \text{ e } X = 12) \neq p(X_1 = 1) \cdot p(X = 12)$. Esse exemplo mostra que X_1 e X não são independentes. ◀

O valor esperado do produto de duas variáveis aleatórias independentes é o produto de seus valores esperados, como mostra o Teorema 5.

TEOREMA 5 Se X e Y são variáveis aleatórias independentes de um espaço S, então $E(XY) = E(X)E(Y)$.

Demonstração: A partir da definição de valor esperado e como X e Y são variáveis aleatórias independentes, temos que

$$\begin{aligned} E(XY) &= \sum_{s \in S} X(s)Y(s)p(s) \\ &= \sum_{r_1 \in X(S), r_2 \in Y(S)} r_1 r_2 \cdot p(X(s) = r_1 \text{ e } Y(s) = r_2) \\ &= \sum_{r_1 \in X(S), r_2 \in Y(S)} r_1 r_2 \cdot p(X(s) = r_1) \cdot p(Y(s) = r_2) \\ &= \left[\sum_{r_1 \in X(S)} r_1 p(X(s) = r_1)\right] \cdot \left[\sum_{r_2 \in Y(S)} r_2 p(Y(s) = r_2)\right] \\ &= E(X)E(Y). \end{aligned}$$

Isso completa a demonstração. ◁

Note que quando X e Y são variáveis aleatórias que não são independentes, podemos concluir que $E(XY) = E(X)E(Y)$, como mostra o Exemplo 13.

EXEMPLO 13 Considere X e Y como variáveis aleatórias que contam o número de caras e o número de coroas quando uma moeda é lançada duas vezes. Como $p(X = 2) = 1/4$, $p(X = 1) = 1/2$ e $p(X = 0) = 1/4$, pelo Teorema 1 temos

$$E(X) = 2 \cdot \frac{1}{4} + 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{4} = 1.$$

Uma computação semelhante mostra que $E(Y) = 1$. Notamos que $XY = 0$ quando duas caras e nenhuma coroa ou duas coroas e nenhuma cara saem, e que $XY = 1$ quando uma cara e uma coroa saem. Assim,

$$E(XY) = 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = \frac{1}{2}.$$

Temos que

$$E(XY) \neq E(X)E(Y).$$

Isso não contradiz o Teorema 5 porque X e Y não são independentes, como o leitor deve verificar (veja o Exercício 16). ◀

Variância

Links

O valor esperado de uma variável aleatória nos diz sobre seu valor médio, mas não nos diz nada sobre como seus valores são amplamente distribuídos. Por exemplo, se X e Y forem variáveis aleatórias do conjunto $S = \{1, 2, 3, 4, 5, 6\}$, com $X(s) = 0$ para todo $s \in S$ e $Y(s) = -1$ se $s \in \{1, 2, 3\}$ e $Y(s) = 1$ se $s \in \{4, 5, 6\}$, então os valores esperados de X e Y são ambos zero. Entretanto, a variável aleatória X nunca varia a partir de 0, enquanto a variável aleatória Y sempre difere de 0 por 1. A variância de uma variável aleatória nos ajuda a caracterizar como uma variável aleatória é amplamente distribuída.

DEFINIÇÃO 4

Considere X como uma variável aleatória no espaço amostral S. A *variância* de X, indicada por $V(X)$, é

$$V(X) = \sum_{s \in S} (X(s) - E(X))^2 p(s).$$

O *desvio-padrão* de X, indicado por $\sigma(X)$, é definido como $\sqrt{V(X)}$.

O Teorema 6 fornece uma expressão simples e útil para a variância de uma variável aleatória.

TEOREMA 6

Se X for uma variável aleatória no espaço amostral S, então $V(X) = E(X^2) - E(X)^2$.

Demonstração: Note que

$$\begin{aligned} V(X) &= \sum_{s \in S} (X(s) - E(X))^2 p(s) \\ &= \sum_{s \in S} X(s)^2 p(s) - 2E(X) \sum_{s \in S} X(s) p(s) + E(X)^2 \sum_{s \in S} p(s) \\ &= E(X^2) - 2E(X)E(X) + E(X)^2 \\ &= E(X^2) - E(X)^2. \end{aligned}$$

Usamos o fato de que $\sum_{s \in S} p(s) = 1$ na penúltima passagem. ◁

EXEMPLO 14 Qual é a variância da variável aleatória X com $X(t) = 1$ se um teste de Bernoulli for um sucesso e $X(t) = 0$ se ele for um fracasso, em que p é a probabilidade de sucesso?

Exemplos Extras

Solução: Como X assume apenas os valores 0 e 1,temos que $X^2(t) = X(t)$. Assim,

$$V(X) = E(X^2) - E(X)^2 = p - p^2 = p(1 - p) = pq.$$ ◀

EXEMPLO 15 **Variância do Valor de um Dado** Qual é a variância de uma variável aleatória X, em que X é o número de resultados quando um dado é lançado?

Solução: Temos $V(X) = E(X^2) - E(X)^2$. Pelo Exemplo 1, sabemos que $E(X) = 7/2$. Para encontrar $E(X^2)$, note que X^2 admite os valores i^2, $i = 1, 2, \ldots, 6$, cada um com probabilidade $1/6$. Temos que

$$E(X^2) = \frac{1}{6}(1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2) = \frac{91}{6}.$$

Concluímos que

$$V(X) = \frac{91}{6} - \left(\frac{7}{2}\right)^2 = \frac{35}{12}.$$ ◀

EXEMPLO 16 Qual é a variância da variável aleatória $X((i, j)) = 2i$, em que i é o número que aparece no primeiro dado e j é o número que aparece no segundo dado, quando dois dados são lançados?

Solução: Usaremos o Teorema 6 para encontrar a variância de X. Para fazer isso, precisamos encontrar os valores esperados de X e de X^2. Note que como $p(X = k)$ é $1/6$ para $k = 2, 4, 6, 8, 10, 12$ e é 0 se ocorrer o contrário,

$$E(X) = (2 + 4 + 6 + 8 + 10 + 12)/6 = 7,$$

e

$$E(X^2) = (2^2 + 4^2 + 6^2 + 8^2 + 10^2 + 12^2)/6 = 182/3.$$

Temos a partir do Teorema 6 que

$$V(X) = E(X^2) - E(X)^2 = 182/3 - 49 = 35/3.$$ ◀

Outro fato útil sobre variâncias é que a variância da soma de duas variáveis aleatórias independentes é a soma de suas respectivas variâncias. Este resultado pode ser usado para computar a variância do resultado de n testes independentes de Bernoulli, por exemplo.

TEOREMA 7 Se X e Y forem duas variáveis aleatórias independentes de um espaço amostral S, então $V(X + Y) = V(X) + V(Y)$. Além disso, se X_i, $i = 1, 2, \ldots, n$, com n um número inteiro positivo, são variáveis aleatórias independentes de S, então $V(X_1 + X_2 + \cdots + X_n) = V(X_1) + V(X_2) + \cdots + V(X_n)$.

Demonstração: A partir do Teorema 6, temos

$$V(X + Y) = E((X + Y)^2) - E(X + Y)^2.$$

Temos que

$$\begin{aligned} V(X + Y) &= E(X^2 + 2XY + Y^2) - (E(X) + E(Y))^2 \\ &= E(X^2) + 2E(XY) + E(Y^2) - E(X)^2 - 2E(X)E(Y) - E(Y)^2. \end{aligned}$$

Como X e Y são independentes, pelo Teorema 5 temos que $E(XY) = E(X)E(Y)$. Vemos que

$$\begin{aligned} V(X + Y) &= (E(X^2) - E(X)^2) + (E(Y^2) - E(Y)^2) \\ &= V(X) + V(Y). \end{aligned}$$

Deixamos a demonstração do caso com n variáveis aleatórias independentes para o leitor (Exercício 28). Tal demonstração pode ser construída generalizando a demonstração que demos para o caso de duas variáveis aleatórias. Note que não é possível usar a indução matemática de forma direta para demonstrar este caso geral (veja o Exercício 27). ◁

EXEMPLO 17 Encontre a variância e o desvio-padrão da variável aleatória X cujo valor, quando dois dados são lançados, é $X((i, j)) = i + j$, em que i é o número que aparece no primeiro dado e j é o número que aparece no segundo dado.

Solução: Considere X_1 e X_2 como variáveis aleatórias definidas por $X_1((i, j)) = i$ e $X_2((i, j)) = j$ para o lançamento de dados. Então, $X = X_1 + X_2$ e X_1 e X_2 são independentes, como mostrou o Exemplo 11. A partir do Teorema 7, temos que $V(X) = V(X_1) + V(X_2)$. Uma computação simples, como no Exemplo 16, juntamente com o Exercício 25 dos Exercícios Complementares no final deste capítulo, nos diz que $V(X_1) = V(X_2) = 35/12$. Assim, $V(X) = 35/12 + 35/12 = 35/6$ e $\sigma(X) = \sqrt{35/6}$. ◀

Vamos agora encontrar a variância da variável aleatória que conta o número de sucessos quando n testes independentes de Bernoulli são realizados.

EXEMPLO 18 Qual é a variância do número de sucessos quando n testes independentes de Bernoulli são realizados, em que p é a probabilidade de sucesso em cada teste?

Solução: Considere X_i como a variável aleatória com $X_i((t_1, t_2, \ldots, t_n)) = 1$ se t_i for um sucesso e $X_i((t_1, t_2, \ldots, t_n)) = 0$ se t_i for um fracasso. Considere $X = X_1 + X_2 + \cdots + X_n$. Então X conta o número de sucessos nos n testes. A partir do Teorema 7, temos que $V(X) = V(X_1) + V(X_2) + \cdots + V(X_n)$. Usando o Exemplo 14, temos $V(X_i) = pq$ para $i = 1, 2, \ldots, n$. Assim, vemos que $V(X) = npq$. ◀

Desigualdade de Chebyshev

Quão provável é uma variável aleatória assumir um valor diferente do seu valor esperado? O Teorema 8, chamado de Desigualdade de Chebyshev, nos ajuda a responder essa questão fornecendo uma grande margem de acerto na probabilidade de o valor de uma variável aleatória diferir do valor esperado.

Links

PAFNUTY LVOVICH CHEBYSHEV (1821–1894) Chebyshev nasceu na aristocracia em Okatovo, na Rússia. Seu pai era um oficial do Exército aposentado que lutou contra Napoleão. Em 1832, a família, com nove filhos, se mudou para Moscou, onde Pafnuty completou o segundo grau em casa. Ele entrou para o Departamento de Física e Matemática da Universidade de Moscou. Como estudante, desenvolveu um novo método de aproximação de raízes de equações. Ele concluiu a graduação na Universidade de Moscou em 1841 com um diploma em matemática e continuou seus estudos, passando no exame para o doutorado em 1843 e defendendo sua tese em 1846.

Chebyshev foi indicado em 1847 para um cargo de assistente na Universidade de São Petersburgo. Ele escreveu e defendeu uma tese em 1847. Tornou-se professor na Universidade de São Petersburgo em 1860, cargo que manteve até 1882.

Seu livro sobre teoria das congruências, escrito em 1849, exerceu grande influência no desenvolvimento da teoria dos números. Seu trabalho sobre distribuição de números primos foi produtivo. Ele provou a conjectura de Bertrand, que para todo número inteiro $n > 3$, há um número primo entre n e $2n - 2$. Chebyshev ajudou a desenvolver as idéias que foram utilizadas posteriormente para demonstrar o Teorema dos Números Primos. O trabalho de Chebyshev sobre aproximação das funções usando polinômios é muito usado quando computações são utilizadas para encontrar os valores das funções. Chebyshev também se interessava por mecânica e estudou a conversão de energia de rotação em energia cinética por acoplagem mecânica. O movimento paralelo de Chebyshev é formado por três barras unidas que aproximam o movimento retilíneo.

TEOREMA 8 **DESIGUALDADE DE CHEBYSHEV** Considere X como uma variável aleatória no espaço amostral S com função de probabilidade p. Se r for um número real positivo, então

$$p(|X(s) - E(X)| \geq r) \leq V(X)/r^2.$$

Demonstração: Considere A como o evento

$$A = \{s \in S \mid |X(s) - E(X)| \geq r\}.$$

O que queremos demonstrar é que $p(A) \leq V(X)/r^2$. Note que

$$\begin{aligned} V(X) &= \sum_{s \in S} (X(s) - E(X))^2 p(s) \\ &= \sum_{s \in A} (X(s) - E(X))^2 p(s) + \sum_{s \notin A} (X(s) - E(X))^2 p(s). \end{aligned}$$

A segunda soma na expressão é não negativa, pois cada uma das suas somatórias é não negativa. Além disso, como para cada elemento s em A, $(X(s) - E(X))^2 \geq r^2$, a primeira soma na expressão é pelo menos $\sum_{s \in A} r^2 p(s)$. Assim, $V(X) \geq \sum_{s \in A} r^2 p(s) = r^2 p(A)$. Daí segue-se que $V(X)/r^2 \geq p(A)$, assim $p(A) \leq V(X)/r^2$, completando a demonstração. ◁

EXEMPLO 19 **Desvios a partir da Média ao Contar Coroas** Suponha que X seja a variável aleatória que conta o número de coroas quando uma moeda não-viciada é jogada n vezes. Note que X é o número de sucessos quando são realizados n testes independentes de Bernoulli, cada um com probabilidade de sucesso de $1/2$. Temos que $E(X) = n/2$ (pelo Teorema 2) e $V(X) = n/4$ (pelo Exemplo 18). Aplicando a Desigualdade de Chebyshev com $r = \sqrt{n}$, mostramos que

$$p(|X(s) - n/2| \geq \sqrt{n}) \leq (n/4)/(\sqrt{n})^2 = 1/4.$$

Conseqüentemente, a probabilidade não é maior que $1/4$ de o número de coroas que aparecem quando uma moeda não-viciada é jogada n vezes desviar da média mais de $\sqrt{n}$. ◀

A Desigualdade de Chebyshev, embora aplicada a qualquer variável aleatória, normalmente falha ao fornecer uma estimativa prática para a probabilidade do valor de uma variável aleatória que excede muito a média. Isso é mostrado no Exemplo 20.

EXEMPLO 20 Considere X como a variável aleatória cujo valor é o número que aparece quando um dado não viciado é lançado. Temos $E(X) = 7/2$ (veja o Exemplo 1) e $V(X) = 35/12$ (veja o Exemplo 15). Como os únicos valores possíveis de X são 1, 2, 3, 4, 5 e 6, X não pode assumir um valor maior que $5/2$ a partir de seu significado, $E(X) = 7/2$. Assim, $p(|X - 7/2| \geq r) = 0$ se $r > 5/2$. Pela Desigualdade de Chebyshev, sabemos que $p(|X - 7/2| \geq r) \leq (35/12)/r^2$. Por exemplo, quando $r = 3$, a Desigualdade de Chebyshev nos diz que $p(|X - 7/2| \geq 3) \leq (35/12)/9 = 35/108$, que é uma estimativa pobre, porque $p(|X - 7/2| \geq 3) = 0$. ◀

Exercícios

1. Qual é o número esperado de caras quando uma moeda não-viciada é jogada cinco vezes?
2. Qual é o número esperado de caras quando uma moeda não-viciada é jogada 10 vezes?
3. Qual é o número esperado de vezes em que aparece um 6 quando um dado não-viciado é lançado 10 vezes?
4. Uma moeda é modificada para que a probabilidade de saírem caras quando ela é lançada seja de 0,6. Qual é o número esperado de caras quando ela é jogada 10 vezes?

5. Qual é a soma esperada dos números que aparecem em dois dados, ambos viciados para que um 3 apareça duas vezes mais que qualquer outro número?

6. Qual é o valor esperado quando é comprado um bilhete de loteria de R$ 1,00 cujo portador ganha R$ 10 milhões se o bilhete tiver os seis números vencedores escolhidos a partir do conjunto {1, 2, 3, . . . , 50} e não ganha nada se o contrário ocorrer?

7. O exame final de um curso de matemática discreta consiste em 50 questões do tipo verdadeiro/falso, cada uma valendo dois pontos, e 25 questões de múltipla escolha, cada uma valendo quatro pontos. A probabilidade de Linda responder a uma questão verdadeiro/falso corretamente é de 0,9, e a probabilidade de ela responder a uma questão de múltipla escolha corretamente é de 0,8. Qual é a nota esperada para ela no final?

8. Qual é a soma esperada dos números que aparecem quando três dados não-viciados são lançados?

9. Suponha que a probabilidade de x estar em uma lista de n números inteiros distintos é de $2/3$ e que é igualmente provável que x seja igual a qualquer elemento da lista. Encontre o número médio de comparações usadas por um algoritmo de busca linear para encontrar x ou para determinar que ele não está na lista.

10. Suponha que joguemos uma moeda até que saia uma coroa duas vezes ou que a joguemos seis vezes. Qual é o número esperado de vezes que jogaremos a moeda?

11. Suponha que lancemos um dado até que um 6 apareça ou que o lancemos 10 vezes. Qual é o número esperado de vezes que jogaremos o dado?

12. Suponha que lancemos um dado até que um 6 apareça.

a) Qual é a probabilidade de jogarmos o dado n vezes?

b) Qual é o número esperado de vezes que jogaremos o dado?

13. Suponha que joguemos um par de dados até que a soma de seus números seja sete. Qual é o número esperado de vezes que lançaremos os dados?

14. Mostre que a soma das probabilidades de uma variável aleatória com distribuição geométrica de parâmetro p, em que $0 < p \leq 1$, é igual a 1.

15. Mostre que se uma variável aleatória X tiver uma distribuição geométrica de parâmetro p, e j for um número inteiro positivo, então $p(X \geq j) = (1 - p)^{j-1}$.

16. Considere X e Y como variáveis aleatórias que contam o número de caras e o número de coroas quando duas moedas são jogadas. Mostre que X e Y não são independentes.

17. Faça a estimativa do número esperado de inteiros com 1000 dígitos que precisam ser selecionados aleatoriamente para se encontrar um número primo, se a probabilidade de um número com 1000 dígitos ser primo é aproximadamente $1/2302$.

18. Suponha que X e Y sejam variáveis aleatórias e que X e Y sejam não negativos para todos os pontos em um espaço amostral S. Considere Z como a variável aleatória definida por $Z(s) = \max(X(s), Y(s))$ para todos os elementos $s \in S$. Mostre que $E(Z) \leq E(X) + E(Y)$.

19. Considere X como o número que aparece no primeiro dado quando dois dados são lançados e considere Y como a soma dos números que aparecem nos dois dados. Mostre que $E(X)E(Y) \neq E(XY)$.

20. Considere A como um evento. Então, I_A, o **indicador de variável aleatória** de A, é igual a 1 se A ocorrer e igual a 0 se não ocorrer. Mostre que a expectativa do indicador de variável aleatória de A é igual à probabilidade de A, ou seja, $E(I_A) = p(A)$.

21. Uma **run** é a maior seqüência de sucessos em uma seqüência de testes de Bernoulli. Por exemplo, na seqüência *S, S, S, F, S, S, F, F, S*, em que S representa sucesso e F representa fracasso, há três seqüências que consistem em três sucessos, dois sucessos e um sucesso, respectivamente. Considere R como a variável aleatória no conjunto de seqüências de n testes independentes de Bernoulli que contam o número de runs nessa seqüência. Encontre $E(R)$. [*Dica:* Mostre que $R = \sum_{j=1}^{n} I_j$, em que $I_j = 1$ se uma seqüência começar no j-ésimo teste de Bernoulli, e $I_j = 0$ se não. Encontre $E(I_1)$ e, então, encontre $E(I_j)$, em que $1 < j \leq n$.]

22. Considere $X(s)$ como uma variável aleatória, em que $X(s)$ é um número inteiro não negativo para todo $s \in S$, e considere A_k como o evento de que $X(s) \geq k$. Mostre que $E(X) = \sum_{k=1}^{\infty} p(A_k)$.

23. Qual é a variância do número de caras quando uma moeda não-viciada é jogada 10 vezes?

24. Qual é a variância do número de vezes que um 6 aparece quando um dado não-viciado é lançado 10 vezes?

25. Considere X_n como a variável aleatória que é igual ao número de coroas menos o número de caras quando n moedas são jogadas.

a) Qual é o valor esperado de X_n?

b) Qual é a variância de X_n?

26. Forneça um exemplo que mostre que a variância da soma de duas variáveis aleatórias não é necessariamente igual à soma de suas variâncias se as variáveis aleatórias não são independentes.

27. Suponha que X_1 e X_2 sejam testes independentes de Bernoulli, cada um com probabilidade 1/2, e que $X_3 = (X_1 + X_2)$ **mod** 2.

a) Mostre que X_1, X_2 e X_3 são independentes, mas X_3 e $X_1 + X_2$ não são independentes.

b) Mostre que $V(X_1 + X_2 + X_3) = V(X_1) + V(X_2) + V(X_3)$.

c) Explique por que uma demonstração por indução matemática do Teorema 7 não funciona, considerando as variáveis aleatórias X_1, X_2 e X_3.

***28.** Prove o caso geral do Teorema 7. Ou seja, mostre que se X_1, X_2, . . . , X_n forem variáveis aleatórias independentes em um espaço amostral S, em que n é um número inteiro positivo, então $V(X_1 + X_2 + \cdots + X_n) = V(X_1) + V(X_2) + \cdots + V(X_n)$. [*Dica:* Generalize a demonstração dada para o Teorema 7 para duas variáveis aleatórias. Note que uma demonstração usando a indução matemática não é válida; veja o Exercício 27.]

29. Use a Desigualdade de Chebyshev para encontrar uma margem maior na probabilidade de que o número de coroas que aparecem quando uma moeda não-viciada é jogada n vezes desvia da média mais que $5\sqrt{n}$.

30. Use a Desigualdade de Chebyshev para encontrar uma margem maior na probabilidade de que o número de coroas

que aparecem quando uma moeda viciada com a probabilidade de caras igual a 0,6 é jogada n vezes desvia da média mais que $\sqrt{n}$.

31. Considere X como uma variável aleatória do espaço amostral S, tal que $X(s) \geq 0$ para todo $s \in S$. Mostre que $p(X(s) \geq a) \leq E(X)/a$ para todo número real positivo a. Essa desigualdade é chamada de **Desigualdade de Markov.**

32. Suponha que o número de latas de refrigerante de limão cheias, em um dia, em uma fábrica de envasamento é uma variável aleatória com um valor esperado de 10 000 e uma variância de 1000.

a) Use a Desigualdade de Markov (Exercício 31) para obter um limite superior na probabilidade de a fábrica encher mais que 11 000 latas em um determinado dia.

b) Use a Desigualdade de Chebyshev para obter um limite inferior da probabilidade de a fábrica encher entre 9 000 e 11 000 latinhas em determinado dia.

33. Suponha que o número de latinhas recicladas, em um dia, em um centro de reciclagem é uma variável aleatória com um valor esperado de 50 000 e uma variância de 10 000.

a) Use a Desigualdade de Markov (Exercício 31) para obter um limite superior na probabilidade de o centro reciclar mais de 55 000 latinhas em um determinado dia.

b) Use a Desigualdade de Chebyshev para obter um limite inferior na probabilidade de o centro reciclar entre 40 000 e 60 000 latinhas em um determinado dia.

***34.** Suponha que a probabilidade de x ser o i-ésimo elemento de uma lista de n números inteiros distintos é $i/[n(n+1)]$. Encontre o número médio de comparações usadas pelo algoritmo de busca linear para encontrar x ou para determinar que ele não está na lista.

***35.** Neste exercício derivamos uma estimativa da complexidade de caso médio para a variância do algoritmo bubble sort que termina quando não há mais trocas a serem feitas. Considere X como a variável aleatória no conjunto de permutações do conjunto de n números inteiros distintos $\{a_1, a_2, \ldots, a_n\}$ com $a_1 < a_2 < \cdots < a_n$, tal que $X(P)$ seja igual ao número de comparações usadas pela ordenação por flutuação para colocar esses inteiros em ordem crescente.

a) Mostre que, considerando a hipótese de que a entrada pode ser qualquer das $n!$ permutações desses inteiros, o número médio de comparações usadas pela ordenação por flutuação é igual a $E(X)$.

b) Use o Exemplo 5 da Seção 3.3 para mostrar que $E(X) \leq n(n-1)/2$.

c) Mostre que a ordenação faz pelo menos uma comparação para toda inversão de dois números inteiros na entrada.

d) Considere $I(P)$ como a variável aleatória que é igual ao número de inversões na permutação P. Mostre que $E(X) \geq E(I)$.

e) Considere $I_{j,k}$ como a variável aleatória com $I_{j,k}(P) = 1$ se a_k preceder a_j em P e $I_{j,k} = 0$ se isso não ocorrer. Mostre que $I(P) = \sum_k \sum_{j<k} I_{j,k}(P)$.

f) Mostre that $E(I) = \sum_k \sum_{j<k} E(I_{j,k})$.

g) Mostre que $E(I_{j,k}) = 1/2$. [*Dica:* Mostre que $E(I_{j,k}) =$ probabilidade de a_k preceder a_j na permutação P. Então, mostre que é igualmente provável para a_k preceder a_j como é para a_j preceder a_k na permutação.]

h) Use os itens (f) e (g) para mostrar que $E(I) = n(n-1)/4$.

i) Conclua a partir dos itens (a), (b) e (h) que o número médio de comparações usadas para organizar n números inteiros é $\Theta(n^2)$.

***36.** Neste exercício, encontramos a complexidade de caso médio do algoritmo de quick sort, descrito no Exercício 50 da Seção 4.4, assumindo uma distribuição uniforme no conjunto de permutações.

a) Considere X como o número de comparações usadas pelo algoritmo de quick sort para organizar uma lista de n números inteiros distintos. Mostre que o número médio de comparações usadas por esse algoritmo é $E(X)$ (em que o espaço amostral é o conjunto de todas as $n!$ permutações de n números inteiros).

b) Considere $I_{j,k}$ como a variável aleatória que é igual a 1 se o j-ésimo menor elemento e o k-ésimo menor elemento da lista inicial forem comparados como o algoritmo de quick sort que ordena a lista, e é igual a 0 se isto não ocorrer. Mostre que $X = \sum_{k=2}^{n} \sum_{j=1}^{k-1} I_{j,k}$.

c) Mostre que $E(X) = \sum_{k=2}^{n} \sum_{j=1}^{k-1} p$ (o j-ésimo menor elemento e o k-ésimo menor elemento são comparados).

d) Mostre que p (o j-ésimo menor elemento e o k-ésimo menor elemento são comparados), em que $k > j$, é igual a $2/(k-j+1)$.

e) Use os itens (c) e (d) para mostrar que $E(X) = 2(n+1)(\sum_{i=2}^{n} 1/i) - 2(n-1)$.

f) Conclua a partir do item (e) e do fato que $\sum_{j=1}^{n} 1/j \approx \ln n + \gamma$, em que $\gamma = 0{,}57721\ldots$ é a constante de Euler, que o número médio de comparações usadas pelo algoritmo de quick sort é $\Theta(n \log n)$.

***37.** Qual é a variância do número de **elementos fixos,** ou seja, os elementos que permanecem na mesma posição, de uma permutação escolhida aleatoriamente de n elementos? [*Dica:* Considere X como o número de pontos fixos da permutação aleatória. Escreva $X = X_1 + X_2 + \cdots + X_n$, em que $X_i = 1$ se a permutação fixa o i-ésimo elemento e $X_i = 0$ caso contrário.]

A **covariância** de duas variáveis aleatórias X e Y de um espaço amostral S, indicada por $\text{Cov}(X, Y)$, é definida como o valor esperado da variável aleatória $(X - E(X))(Y - E(Y))$. Ou seja, $\text{Cov}(X, Y) = E((X - E(X))(Y - E(Y)))$.

38. Mostre que $\text{Cov}(X, Y) = E(XY) - E(X)E(Y)$ e use este resultado para concluir que $\text{Cov}(X, Y) = 0$ se X e Y forem variáveis aleatórias independentes.

39. Mostre que $V(X+Y) = V(X) + V(Y) + 2\,\text{Cov}(X, Y)$.

40. Encontre $\text{Cov}(X, Y)$ se X e Y forem variáveis aleatórias com $X((i, j)) = 2i$ e $Y((i, j)) = i + j$, em que i e j são os números que aparecem no primeiro e no segundo dados quando eles são lançados.

41. Quando m bolas são distribuídas em n caixas uniforme e aleatoriamente, qual é a probabilidade de a primeira caixa permanecer vazia?

42. Qual é o número esperado de bolas que aparecem na primeira caixa quando m bolas são distribuídas em n caixas aleatória e uniformemente?

43. Qual é o número esperado de caixas que permanecem vazias quando m bolas são distribuídas em n caixas aleatória e uniformemente?

Termos-chave e Resultados

TERMOS

espaço amostral: o conjunto de possíveis resultados de um experimento

evento: um subconjunto do espaço amostral de um experimento

probabilidade de um evento (definição de Laplace): o número de resultados que tiveram sucesso nesse evento dividido pelo número de resultados possíveis

distribuição de probabilidade: uma função p do conjunto de todos os resultados de um espaço amostral S para a qual $0 \leq p(x_i) \leq 1$ para $i = 1, 2, \ldots, n$ e $\sum_{i=1}^{n} p(x_i) = 1$, em que x_i são os possíveis resultados

probabilidade de um evento E: a soma das probabilidades dos resultados em E

$p(E|F)$ (probabilidade condicional de E dado F): a razão $p(E \cap F)/p(F)$

eventos independentes: os eventos E e F, tal que $p(E \cap F) = p(E)p(F)$

variável aleatória: uma função que determina um número real para cada resultado de um experimento

distribuição de uma variável aleatória X: o conjunto de pares $(r, p(X = r))$ para $r \in X(S)$

distribuição uniforme: a designação de probabilidades iguais para os elementos de um conjunto finito

valor esperado de uma variável aleatória: a média de uma variável aleatória, com valores dessa variável pela probabilidade de resultados, ou seja, $E(X) = \sum_{s \in S} p(s)X(s)$

distribuição geométrica: a distribuição de uma variável aleatória X tal que $p(X = k) = (1 - p)^{k-1} p$ para $k = 1, 2, \ldots$ para qualquer p

variáveis aleatórias independentes: as variáveis aleatórias X e Y, tal que $p(X(s) = r_1 \text{ e } Y(s) = r_2) = p(X(s) = r_1)p(Y(s) = r_2)$ para todos os números reais r_1 e r_2

variância de uma variável aleatória: a média do quadrado da diferença entre o valor dado e a probabilidade de resultados, ou seja, $V(X) = \sum_{s \in S} (X(s) - E(X))^2 p(s)$

teste de Bernoulli: um experimento com dois resultados possíveis

algoritmo probabilístico (ou Monte Carlo): um algoritmo no qual escolhas aleatórias são feitas em uma ou mais etapas

método probabilístico: uma técnica para demonstrar a existência de objetos em um conjunto com determinadas propriedades que funciona determinando as probabilidades dos objetos e mostrando que a probabilidade de um objeto ter essas propriedades é positiva

RESULTADOS

A probabilidade de k sucessos quando n testes independentes de Bernoulli são realizados é igual a $C(n, k)p^k q^{n-k}$, em que p é a probabilidade de sucesso e $q = 1 - p$ é a probabilidade de fracasso.

Teorema de Bayes: Se E e F são eventos de um espaço amostral S, tal que $p(E) \neq 0$ e $p(F) \neq 0$, então

$$p(F \mid E) = \frac{p(E \mid F)p(F)}{p(E \mid F)p(F) + p(E \mid \overline{F})p(\overline{F})}.$$

$$E(X) = \sum_{r \in X(S)} p(X = r)r.$$

linearidade da esperança: $E(X_1 + X_2 + \cdots + X_n) = E(X_1) + E(X_2) + \cdots + E(X_n)$ se $X_1, X_2, \ldots, X_n$ são variáveis aleatórias.

Se X e Y forem variáveis aleatórias independentes, então $E(XY) = E(X)E(Y)$.

Se $X_1, X_2, \ldots, X_n$ forem variáveis aleatórias independentes, então $V(X_1 + X_2 + \cdots + X_n) = V(X_1) + V(X_2) + \cdots + V(X_n)$.

Desigualdade de Chebyshev: $p(|X(s) - E(X)| \geq r) \leq V(X)/r^2$, em que X é uma variável aleatória com função probabilística p, e r é um número real positivo.

Questões de Revisão

1. **a)** Defina a probabilidade de um evento quando todos os resultados são igualmente prováveis.
b) Qual é a probabilidade de você selecionar seis números vencedores em uma loteria se os seis diferentes números são escolhidos a partir dos primeiros 50 números inteiros positivos?

2. **a)** Quais as condições que devem ser levadas em conta pelas probabilidades designadas aos resultados de um espaço amostral finito?
b) Quais probabilidades devem ser designadas aos resultados de cara e de coroa se cara aparece três vezes mais que coroa?

3. **a)** Defina a probabilidade condicional de um evento E dado um evento F.
b) Suponha que E seja o evento que consta de um dado ser lançado e aparecer um número par e F seja o evento que consta de um dado ser lançado e aparecer 1, 2 ou 3. Qual é a probabilidade de F dado E?

4. **a)** Quando dois eventos E e F são independentes?
b) Suponha que E seja o evento de aparecer um número par quando um dado não-viciado é lançado e F seja o evento de sair 5 ou 6. E e F são independentes?

5. **a)** O que é uma variável aleatória?
b) Quais são os valores designados a uma variável aleatória X que determina a cada lançamento de dois dados o maior número dos dois?

6. **a)** Defina o valor esperado de uma variável aleatória X.
b) Qual é o valor esperado da variável aleatória X que designa a um lançamento de dois dados o maior número dos dois?

7. **a)** Explique como a complexidade computacional de caso médio de um algoritmo, com valores de entrada finitos, pode ser interpretada como um valor esperado.

b) Qual é a complexidade computacional de caso médio do algoritmo de busca linear, se a probabilidade do elemento que é buscado na lista é $1/3$, e é igualmente provável que esse elemento seja qualquer dos n elementos da lista?

8. a) O que você entende por teste de Bernoulli?

b) Qual é a probabilidade de k sucessos em n testes independentes de Bernoulli?

c) Qual é o valor esperado do número de sucessos em n testes independentes de Bernoulli?

9. a) O que é a linearidade das esperanças?

b) Como a linearidade das esperanças pode nos ajudar a encontrar o número esperado de pessoas que recebem o chapéu certo quando um guardador devolve os chapéus aleatoriamente a cada pessoa?

10. a) Como a probabilidade pode ser usada para resolver problemas que implicam decisão, se uma pequena possibilidade de erro é aceita?

b) Como podemos determinar rapidamente se um número inteiro positivo é primo, se aceitamos uma pequena possibilidade de erro?

11. Enuncie o Teorema de Bayes e use-o para encontrar $p(F \mid E)$ se $p(E \mid F) = 1/3$, $p(E \mid \overline{F}) = 1/4$ e $p(F) = 2/3$, em que E e F são eventos de um espaço amostral S.

12. a) O que significa dizermos que uma variável aleatória tem uma distribuição geométrica de parâmetro p?

b) O que significa uma distribuição geométrica de parâmetro p?

13. a) O que é a variância de uma variável aleatória?

b) Qual é a variância de um teste de Bernoulli com probabilidade p de sucesso?

14. a) Qual é a variância da soma das n variáveis aleatórias independentes?

b) Qual é a variância do número de sucessos quando n testes independentes de Bernoulli, com probabilidade p de sucesso, são realizados?

15. O que a Desigualdade de Chebyshev nos diz sobre a probabilidade de uma variável aleatória ter desvio da média além de um determinado valor?

Exercícios Complementares

1. Qual é a probabilidade de seis números consecutivos serem escolhidos como os números corretos em uma loteria, em que cada número escolhido está entre 1 e 40 (inclusive)?

2. Qual é a probabilidade de uma mão de 13 cartas não conter pares?

3. Qual é a probabilidade de uma mão de bridge com 13 cartas conter

a) todas as 13 de copas?

b) 13 cartas do mesmo naipe?

c) sete espadas e seis paus?

d) sete cartas de um naipe e seis cartas de outro naipe?

e) quatro ouros, seis copas, duas espadas e um paus?

f) quatro cartas de um mesmo naipe, seis cartas de um segundo naipe, duas cartas de um terceiro naipe e uma carta de um quarto naipe?

4. Qual é a probabilidade de uma mão de pôquer com sete cartas conter

a) quatro cartas de um tipo e três cartas de outro?

b) três cartas de um tipo e dois pares de tipos diferentes?

c) pares de cada três tipos diferentes e uma única carta de um quarto tipo?

d) pares de dois tipos diferentes e três cartas de um terceiro, quarto e quinto tipo?

e) cartas de sete tipos diferentes?

f) um *flush* de sete cartas?

g) um *straight* de sete cartas?

h) um *straight flush* de sete cartas?

Um **dado octaedro** tem oito faces numeradas de 1 a 8.

5. a) Qual é o valor esperado do número que aparece quando um dado octaedro não-viciado é lançado?

b) Qual é a variância do número que aparece quando um dado octaedro não-viciado é lançado?

Um **dado dodecaedro** tem 12 faces numeradas de 1 a 12.

6. a) Qual é o valor esperado do número que aparece quando um dado dodecaedro não-viciado é lançado?

b) Qual é a variância do número que aparece quando um dado dodecaedro não-viciado é lançado?

7. Suponha que um par de dados octaedros não-viciados seja lançado.

a) Qual é o valor esperado da soma dos números que aparecem?

b) Qual é a variância da soma dos números que aparecem?

8. Suponha que um par de dados dodecaedros não-viciados seja lançado.

a) Qual é o valor esperado da soma dos números que aparecem?

b) Qual é a variância da soma dos números que aparecem?

9. Suponha que um dado comum não-viciado e um dado octaedro também não-viciado sejam lançados juntos.

a) Qual é o valor esperado da soma dos números que aparecem?

b) Qual é a variância da soma dos números que aparecem?

10. Suponha que um dado octaedro não-viciado e um dado dodecaedro também não-viciado sejam lançados juntos.

a) Qual é o valor esperado da soma dos números que aparecem?

b) Qual é a variância da soma dos números que aparecem?

11. Suponha que n pessoas, $n \geq 3$, joguem "dois ou um" com moedas para decidir quem irá comprar a próxima rodada de refrigerantes. Cada uma das n pessoas lança uma moeda não-viciada simultaneamente. Se todas as moedas, menos uma, forem iguais, a pessoa com resultado diferente compra os refrigerantes. Se não, as pessoas jogam as moedas novamente e continuam até que apenas uma moeda seja diferente das outras.

a) Qual é a probabilidade de a brincadeira ser decidida no primeiro lançamento?

b) Qual é a probabilidade de a brincadeira ser decidida no k-ésimo lançamento?

c) Qual é o número esperado de lançamentos necessários para decidir a brincadeira com n pessoas?

12. Suponha que p e q sejam números primos e $n = pq$. Qual é a probabilidade de se escolher aleatoriamente um número inteiro positivo menor que n que não seja divisível por p nem por q?

***13.** Suponha que m e n sejam números inteiros positivos. Qual é a probabilidade de se escolher aleatoriamente um número inteiro positivo menor que mn que não seja divisível por m nem por n?

14. Suponha que $E_1, E_2, \ldots, E_n$ sejam n eventos com $p(E_i) > 0$ para $i = 1, 2, \ldots, n$. Mostre que

$$p(E_1 \cap E_2 \cap \cdots \cap E_n) = p(E_1)\, p(E_2 \mid E_1)\, p(E_3 \mid E_1 \cap E_2) \cdots p(E_n \mid E_1 \cap E_2 \cap \cdots \cap E_{n-1}).$$

15. Há três cartões em uma caixa. Ambos os lados de um cartão são pretos, ambos os lados de outro são vermelhos e o terceiro cartão tem um lado preto e um vermelho. Escolhemos um cartão aleatoriamente e observamos um lado apenas.

a) Se o lado for preto, qual é a probabilidade de o outro lado ser preto?

b) Qual é a probabilidade de o lado oposto ser da mesma cor do lado que observamos?

16. Qual é a probabilidade, quando uma moeda não-viciada é lançada n vezes, de sair um número igual de caras e coroas?

17. Qual é a probabilidade de uma cadeia de bits de extensão 10 escolhida aleatoriamente ser um palíndromo?

18. Qual é a probabilidade de uma cadeia de bits de extensão 11 escolhida aleatoriamente ser um palíndromo?

19. Considere o seguinte jogo. Uma pessoa lança uma moeda repetidamente até que uma cara apareça. Essa pessoa recebe um valor de 2^n reais se a primeira cara aparecer no n-ésimo lançamento.

a) Considere X como uma variável aleatória que é igual à quantidade de dinheiro que a pessoa recebe. Mostre que o valor esperado de X não existe (ou seja, é infinito). Mostre que um apostador racional, ou seja, alguém que paga para jogar um jogo mesmo que o preço para jogar não seja mais que a eliminatória final, deve aceitar apostar qualquer quantia para jogar esse jogo. (Isto é conhecido como **Paradoxo de São Petersburgo.** Por que você acha que ele é chamado de paradoxo?)

b) Suponha que a pessoa receba 2^n reais se a primeira cara aparecer no n-ésimo lançamento, em que $n < 8$ e $2^8 = 256$ reais se a primeira cara sair no oitavo lançamento ou depois dele. Qual é o valor esperado da quantia de dinheiro que essa pessoa recebe? Quanto a pessoa deveria pagar para participar desse jogo?

20. Suponha que n bolas sejam colocadas em b urnas, sendo que é igualmente provável para cada bola cair em qualquer urna e que os lançamentos são independentes.

a) Encontre a probabilidade de determinada bola cair em uma determinada urna.

b) Qual é o número esperado de bolas em determinada urna?

c) Qual o número esperado de lançamentos até que determinada urna contenha uma bola?

***d)** Qual é o número esperado de lançamentos até que todas as urnas contenham uma bola? [*Dica:* Considere X_i como o número de lançamentos necessários para uma bola cair na i-ésima urna, uma vez que $i - 1$ urnas contenham uma bola. Encontre $E(Xi)$ e use a linearidade das esperanças.]

21. Suponha que A e B sejam eventos com probabilidades $p(A) = 3/4$ e $p(B) = 1/3$.

a) Qual pode ser a maior $p(A \cap B)$? E qual a menor? Dê exemplos para mostrar que ambos os extremos para $p(A \cap B)$ são possíveis.

b) Qual pode ser a maior $p(A \cup B)$? E qual a menor? Dê exemplos para mostrar que ambos os extremos para $p(A \cup B)$ são possíveis.

22. Suponha que A e B sejam eventos com probabilidades $p(A) = 2/3$ e $p(B) = 1/2$.

a) Qual pode ser a maior $p(A \cap B)$? E qual a menor? Dê exemplos para mostrar que ambos os extremos para $p(A \cap B)$ são possíveis.

b) Qual é a maior $p(A \cup B)$? E qual é a menor? Dê exemplos para mostrar que ambos os extremos para $p(A \cup B)$ são possíveis.

23. Dizemos que os eventos $E_1, E_2, \cdots, E_n$ são **mutuamente independentes** se $p(E_{i_1} \cap E_{i_2} \cap \cdots \cap E_{i_m}) = p(E_{i_1}) p(E_{i_2}) \cdots p(E_{i_m})$ sempre que $i_j, j = 1, 2, \cdots, m$, forem números inteiros com $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ e $m \geq 2$.

a) Desenvolva as condições necessárias para três eventos, E_1, E_2 e E_3, serem mutuamente independentes.

b) Considere E_1, E_2 e E_3 como os eventos que têm o primeiro lançamento com caras como resultado, o segundo lançamento com coroas como resultado e o terceiro lançamento com coroas como resultado, respectivamente, quando uma moeda não-viciada é lançada três vezes. Os eventos E_1, E_2 e E_3 são mutuamente independentes?

c) Considere E_1, E_2 e E_3 como os eventos em que no primeiro lançamento aparecem caras, no terceiro aparecem caras e aparece um número par de caras, respectivamente, quando uma moeda não viciada é lançada três vezes. Os eventos E_1, E_2 e E_3 são independentes? Eles são mutuamente independentes?

d) Considere E_1, E_2 e E_3 como os eventos em que no primeiro lançamento aparecem caras, no terceiro lançamento aparecem caras e em que em um do primeiro e do terceiro lançamento aparecem caras, respectivamente, quando uma moeda não viciada é lançada três vezes. Os eventos E_1, E_2 e E_3 são independentes? Eles são mutuamente independentes?

e) Quantas condições devem ser checadas para mostrar que n eventos são mutuamente independentes?

24. Suponha que A e B sejam eventos de um espaço amostral S tal que $p(A) \neq 0$ e $p(B) \neq 0$. Mostre que se $p(B \mid A) < p(B)$, então $p(A \mid B) < p(A)$.

25. Considere X como uma variável aleatória de um espaço amostral S. Mostre que $V(aX + b) = a^2V(X)$ sempre que a e b são números reais.

26. Use a Desigualdade de Chebyshev para mostrar que a probabilidade de mais de 10 pessoas pegarem o chapéu correto quando um guardador devolve aleatoriamente os chapéus não excede $1/100$, não importando quantas pessoas chequem seus chapéus. [*Dica:* Use o Exemplo 6 e o Exercício 37 da Seção 6.4.]

27. Suponha que pelo menos um dos eventos E_j, $j = 1, 2, \ldots, m$, é certo de acontecer e não mais que dois podem ocorrer. Mostre que se $p(E_j) = q$ para $j = 1, 2, \ldots, m$ e $p(E_j \cap E_k) = r$ para $1 \leq j < k \leq m$, então $q \geq 1/m$ e $r \leq 2/m$.

28. Mostre que se m for um número inteiro positivo, então a probabilidade do m-ésimo sucesso acontecer no $(m + n)$-ésimo teste quando os testes independentes de Bernoulli, cada um com probabilidade p de sucesso, são executados é $\binom{n+m-1}{n} q^n p^m$.

29. Há n tipos diferentes de cartas que você pode conseguir como prêmio quando compra determinado produto. Suponha que toda vez que você comprar esse produto seja igualmente provável conseguir qualquer uma dessas cartas. Considere X como a variável aleatória igual ao número dos produtos que você precisa comprar para obter pelo menos um tipo de cada carta, e considere X_j como a variável aleatória igual ao número de produtos adicionais que você deve comprar depois que j cartas diferentes forem ganhas até que uma nova carta seja obtida para $j = 0, 1, \ldots, n - 1$.

a) Mostre que $X = \sum_{j=0}^{n-1} X_j$.

b) Mostre que depois que j tipos diferentes de cartas foram obtidos, a carta da próxima compra será a carta de um novo tipo com probabilidade $(n - j)/n$.

c) Mostre que X_j tem uma distribuição geométrica de parâmetro $(n - j)/n$.

d) Use os itens (a) e (c) para mostrar que $E(X) = n\sum_{j=1}^{n} 1/j$.

e) Use a aproximação $\sum_{j=1}^{n} 1/j \approx \ln n + \gamma$, em que $\gamma = 0{,}57721\ldots$ é a constante de Euler, para encontrar o número esperado de produtos que você precisa comprar para conseguir uma carta de cada tipo se existem 50 tipos diferentes de cartas.

30. O **problema de máxima satisfabilidade** pede por uma designação de valores-verdade para as variáveis em uma proposição composta em uma forma normal conjuntiva (que expressa uma proposição composta como a conjunção de sentenças, em que cada sentença é a disjunção de duas ou mais variáveis ou suas negações) que tornam verdadeiras quantas sentenças forem possíveis. Por exemplo, três, mas não quatro, das sentenças em

$$(p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r)$$

podem ser verdadeiras para determinados valores-verdade para p, q e r. Mostraremos que os métodos probabilísticos podem fornecer um limite inferior para o número de sentenças que podem ser verdadeiras com a designação de valores-verdade para as variáveis.

a) Suponha que há n variáveis em uma proposição composta na forma normal conjuntiva. Se escolhermos um valor-verdade para cada variável aleatoriamente, jogando uma moeda e determinando verdadeiro para a variável se virar cara, e falso se virar coroa, qual é a probabilidade de cada possível determinação de valores-verdade para as n variáveis?

b) Assumindo que cada sentença é a disjunção de duas variáveis distintas ou suas negações, qual é a probabilidade de uma determinada sentença ser verdadeira, dada a determinação aleatória de valores-verdade do item (a)?

c) Suponha que haja D sentenças na proposição composta. Qual é o número esperado de sentenças verdadeiras, dada uma determinação aleatória de valores-verdade das variáveis?

d) Use o item (c) para mostrar que para toda proposição composta na forma normal conjuntiva existe uma determinação de valores-verdade para as variáveis que têm pelo menos 3/4 de sentenças verdadeiras.

31. Qual é a probabilidade de cada jogador ter uma mão com um ás quando as 52 cartas de um baralho são distribuídas entre quatro jogadores?

***32.** O método a seguir pode ser usado para construir permutações aleatórias de uma seqüência de n termos. Primeiro, troque o n-ésimo termo com o $r(n)$-ésimo termo, em que $r(n)$ é um número inteiro selecionado aleatoriamente com $1 \leq r(n) \leq n$. Depois, troque o $(n - 1)$-ésimo termo da seqüência resultante com seu $r(n - 1)$-ésimo termo, em que $r(n - 1)$ é um número inteiro selecionado aleatoriamente com $1 \leq r(n - 1) \leq n - 1$. Continue este processo até que $j = n$, em que na j-ésima etapa você troque o $(n - j + 1)$-ésimo termo da seqüência resultante com seu $r(n - j + 1)$-ésimo termo, em que $r(n - j + 1)$ é um número inteiro selecionado aleatoriamente com $1 \leq r(n - j + 1) \leq n - j + 1$. Mostre que quando este método é seguido, cada uma das $n!$ permutações diferentes dos termos da seqüência é igualmente provável de ser construída. [*Dica:* Use a indução matemática, assumindo que a probabilidade de cada uma das permutações dos $n - 1$ termos produzidos por este procedimento para a seqüência de $n - 1$ termos é $1/(n - 1)!$.]

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dado um número real p com $0 \leq p \leq 1$, gere números aleatórios a partir da distribuição de Bernoulli com probabilidade p.
2. Dado um número inteiro positivo n, construa uma permutação aleatória do conjunto $\{1, 2, 3, \ldots, n\}$. (Veja o Exercício 32 dos Exercícios Complementares no final deste capítulo.)
3. Dados os números inteiros positivos m e n, construa m permutações aleatórias dos primeiros n números inteiros positivos. Encontre o número de inversões em cada permutação e determine o número médio dessas inversões.
4. Dado um número inteiro positivo n, simule n jogadas repetidas de uma moeda viciada com probabilidade p de caras e determine o número de caras que aparecem. Mostre os resultados acumulados.
5. Dados os números inteiros positivos n e m, construa m permutações aleatórias dos primeiros n números inteiros positivos. Ordene cada permutação usando a insertion sort, contando o número de comparações usadas. Determine o número médio de comparações usadas nas m permutações.
6. Dados os números inteiros positivos n e m, construa m permutações aleatórias dos primeiros n números inteiros positivos. Ordene cada permutação usando a versão da bubble sort que termina quando um "loop" for feito sem trocas, contando o número de comparações usadas. Determine o número médio de comparações usadas nas m permutações.
7. Dado um número inteiro positivo m, simule a coleção de cartas que vem na compra de produtos para encontrar o número de produtos que devem ser comprados para obter um conjunto completo de m cartas diferentes. (Veja o Exercício Complementar 29.)
8. Dados os números inteiros positivos m e n, simule o ordenamento de n chaves, em que um registro com a chave k é colocado na localização $h(k) = k \bmod m$ e determine se há pelo menos uma colisão.
9. Dado um número inteiro positivo n, encontre a probabilidade de escolher seis números inteiros do conjunto $\{1, 2, \ldots, n\}$ que são selecionados mecanicamente em uma loteria.
10. Simule testes repetidos do problema das três portas de Monty Hall (Exemplo 10 da Seção 6.1) para calcular a probabilidade de vencer com cada estratégia.
11. Dada uma lista de palavras e as probabilidades empíricas de elas ocorrerem em e-mails que são spam e em e-mails que não são spam, determine a probabilidade de uma nova mensagem de e-mail ser spam.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Encontre as probabilidades de cada tipo de mão de pôquer com cinco cartas e classifique esses tipos de mãos a partir de suas probabilidades.
2. Encontre algumas condições para que o valor esperado de compra de um bilhete de loteria de R$ 1,00 em uma loteria tenha um valor esperado de mais de R$ 1,00. Para vencer, você deve escolher seis números, em que a ordem não é relevante, a partir dos números inteiros positivos de 1 a 49, inclusive. Os vencedores são escolhidos eventualmente entre os proprietários dos bilhetes premiados. Tenha certeza de considerar o total do dinheiro do prêmio e o número de pessoas que compram os bilhetes.
3. Faça a estimativa da probabilidade de dois números inteiros escolhidos aleatoriamente serem primos entre si testando um grande número de pares de inteiros escolhidos aleatoriamente. Procure por um teorema que dê essa probabilidade e compare seus resultados com a probabilidade correta.
4. Determine o número de pessoas necessárias para assegurar que a probabilidade de, pelo menos, duas delas fazerem aniversário no mesmo dia do ano seja pelo menos 70%, 80%, 90%, 95%, 98% e 99%.
5. Construa uma lista de 100 permutações escolhidas aleatoriamente do conjunto dos primeiros 100 números inteiros positivos. (Veja o Exercício 32 dos Exercícios Complementares no final deste capítulo.)
6. Dado um grupo de e-mails, cada um determinado como sendo ou não spam, desenvolva um filtro bayesiano baseado na ocorrência de determinadas palavras nessas mensagens.
7. Simule um procedimento "dois ou um" (descrito no Exercício 11 dos Exercícios Complementares) para n pessoas com $3 \leq n \leq 10$. Execute um grande número de testes para cada valor de n e use os resultados para estimar o número esperado de jogadas necessárias. O seu resultado é igual ao encontrado no Exercício 29 da Seção 6.2? Varie o problema supondo que exatamente uma pessoa está com uma moeda viciada com probabilidade de caras $p \neq 0{,}5$.
8. Dado um número inteiro positivo n, simule que uma pessoa devolva aleatoriamente os chapéus. Determine o número de pessoas que pegam o chapéu correto.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Descreva as origens da teoria da probabilidade e os primeiros usos dessa teoria.
2. Descreva as diferentes apostas que você pode fazer quando joga em uma roleta. Encontre a probabilidade de cada uma dessas apostas na versão americana, em que a roleta tem os números 0 e 00. Qual é a melhor aposta e qual é a pior para você?
3. Discuta a probabilidade de ganhar quando você joga vinte um contra um cassino. Há uma estratégia para a pessoa que joga contra a casa ganhar?
4. Investigue o jogo de dados e discuta a probabilidade de o lançador vencer e quão perto está de um jogo honesto.
5. Discuta os assuntos que envolvem o desenvolvimento bem-sucedido de filtros de spam e a situação atual da guerra entre os criadores de spam e as pessoas que tentam acabar com eles.
6. Discuta a história e a solução do que é conhecido como Problema de Newton–Pepys, que pergunta o que é mais provável: sair pelo menos um seis quando seis dados são lançados, sair pelo menos dois seis quando 12 dados são lançados ou sair pelo menos três seis quando 18 dados são lançados.
7. Explique como Erdös e Rényi usaram o primeiro método probabilístico e descreva algumas outras aplicações desse método.
8. Discuta os diferentes tipos de algoritmos probabilísticos e descreva alguns exemplos de cada tipo.

CAPÍTULO

7 Técnicas Avançadas de Contagem

7.1 Relações de Recorrência
7.2 Resolvendo Relações de Recorrência Lineares
7.3 Algoritmos de Divisão e Resolução e Relações de Recorrência
7.4 Funções Generalizadas
7.5 Inclusão–Exclusão
7.6 Aplicações da Inclusão–Exclusão

Muitos problemas de contagem não podem ser resolvidos facilmente utilizando os métodos discutidos no Capítulo 5. Um desses problemas é: Quantas cadeias de bits de extensão n não contêm dois zeros consecutivos? Para resolver esse problema, considere a_n como o número de cadeias de extensão n. Um argumento pode ser dado que mostre que $a_{n+1} = a_n + a_{n-1}$. Esta equação é chamada de relação de recorrência e as condições iniciais $a_1 = 2$ e $a_2 = 3$ determinam a seqüência $\{a_n\}$. Além disso, uma fórmula explícita pode ser encontrada para a_n a partir da equação que relaciona os termos da seqüência. Como veremos, uma técnica semelhante pode ser usada para resolver muitos tipos de problemas de contagem.

Veremos também que muitos problemas de contagem podem ser resolvidos utilizando séries de potências, chamadas de funções geradas, nas quais os coeficientes de potência x representam os termos da seqüência em que estamos interessados. Além de resolver problemas de contagem, seremos também capazes de usar a geração de funções para resolver relações de recorrência e para demonstrar identidades combinatórias.

Muitos outros tipos de problemas de contagem não podem ser resolvidos usando as técnicas discutidas no Capítulo 5, tais como: Quantas maneiras de determinar sete funções a três empregados, para que cada empregado exerça pelo menos uma função, são possíveis? Quantos números primos existem menores que 1 000? Ambos podem ser resolvidos contando o número de elementos da união dos conjuntos. Vamos desenvolver uma técnica, chamada de princípio da inclusão-exclusão, que conta o número de elementos da união de conjuntos, e mostraremos como esse princípio pode ser usado para resolver problemas de contagem.

As técnicas estudadas neste capítulo, ao lado das técnicas básicas do Capítulo 5, podem ser usadas para resolver muitos problemas de contagem.

7.1 Relações de Recorrência

Introdução

O número de bactérias em uma colônia dobra a cada hora. Se uma colônia se inicia com cinco bactérias, quantas existirão em n horas? Para resolver este problema, considere a_n como o número de bactérias ao final das n horas. Como o número de bactérias dobra a cada hora, a relação $a_n = 2a_{n-1}$ é mantida sempre que n for um número inteiro positivo. Essa relação, juntamente com a condição inicial $a_0 = 5$, determina unicamente a_n para todos os números inteiros não negativos n. Podemos encontrar uma fórmula para a_n a partir dessa informação.

Alguns dos problemas de contagem que não podem ser resolvidos com as técnicas descritas no Capítulo 5, podem ser resolvidos encontrando relações, chamadas de relações de recorrência, entre os termos de uma seqüência, como foi feito com o problema que envolve bactérias. Estudaremos uma variedade de problemas de contagem que podem ser moldados usando relações de recorrência. Vamos desenvolver, nesta seção e na Seção 7.2, métodos para encontrar fórmulas explícitas para os termos da seqüência que satisfazem determinados tipos de relações de recorrência.

Relações de Recorrência

Auto-avaliação

No Capítulo 4, discutimos como as seqüências podem ser definidas recursivamente. Lembre-se que a definição recursiva de uma seqüência especifica um ou mais termos iniciais e uma regra para determinar os termos subseqüentes a partir daqueles que os precedem. Uma regra deste úl-

timo tipo (se for ou não parte de uma definição recursiva) é chamada de **relação de recorrência.** Essas relações podem ser usadas no estudo e na resolução de problemas de contagem.

DEFINIÇÃO 1 Uma *relação de recorrência* para a seqüência $\{a_n\}$ é uma equação que expressa a_n em termos de um ou mais termos prévios da seqüência, ou seja, $a_0, a_1, \ldots, a_{n-1}$, para todos os números inteiros n com $n \geq n_0$, em que n_0 é um número inteiro não negativo. Uma seqüência é chamada de *solução* de uma relação de recorrência se seus termos satisfizerem a relação de recorrência.

Há uma importante conexão entre as relações de recursão e as de recorrência as quais exploraremos mais à frente neste capítulo. Um algoritmo recursivo fornece a solução de um problema de tamanho n em termos das soluções de uma ou mais instâncias do mesmo problema em tamanho menor. Conseqüentemente, quando analisamos a complexidade de um algoritmo recursivo, obtemos uma relação de recorrência que expressa o número de operações necessárias para resolver um problema de tamanho n em termos do número de operações necessárias para resolver o problema para uma ou mais instâncias de tamanho menor.

EXEMPLO 1 Considere $\{a_n\}$ como uma seqüência que satisfaz a relação de recorrência $a_n = a_{n-1} - a_{n-2}$ para $n = 2, 3, 4, \ldots$, e suponha que $a_0 = 3$ e $a_1 = 5$. Quais os valores de a_2 e a_3?

Solução: Vemos, a partir da relação de recorrência, que $a_2 = a_1 - a_0 = 5 - 3 = 2$ e $a_3 = a_2 - a_1 = 2 - 5 = -3$. Podemos encontrar a_4, a_5 e cada termo subseqüente da mesma forma. ◀

EXEMPLO 2 Determine se a seqüência $\{a_n\}$, em que $a_n = 3n$ para todo número inteiro não negativo n, é uma solução da relação de recorrência $a_n = 2a_{n-1} - a_{n-2}$ para $n = 2, 3, 4, \ldots$. Responda a mesma questão para $a_n = 2^n$ e $a_n = 5$.

Solução: Suponha que $a_n = 3n$ para todo número inteiro não negativo n. Então, para $n \geq 2$, vemos que $2a_{n-1} - a_{n-2} = 2[3(n-1)] - 3(n-2) = 3n = a_n$. Assim, $\{a_n\}$, em que $a_n = 3n$, é a solução da relação de recorrência.

Suponha que $a_n = 2^n$ para todo número inteiro não negativo n. Note que $a_0 = 1$, $a_1 = 2$ e $a_2 = 4$. Como $2a_1 - a_0 = 2 \cdot 2 - 1 = 3 \neq a_2$, vemos que $\{a_n\}$, em que $a_n = 2^n$, não é uma solução da relação de recorrência.

Suponha que $a_n = 5$ para todo número inteiro não negativo n. Então, para $n \geq 2$, vemos que $a_n = 2a_{n-1} - a_{n-2} = 2 \cdot 5 - 5 = 5 = a_n$. Assim $\{a_n\}$, em que $a_n = 5$, é uma solução para a relação de recorrência. ◀

As **condições iniciais** para uma seqüência especificam os termos que precedem o primeiro termo em que a relação de recorrência surte efeito. Como ocorre no Exemplo 1, $a_0 = 3$ e $a_1 = 5$ são as condições iniciais. A relação de recorrência e as condições iniciais determinam unicamente uma seqüência. Este é o caso, pois uma relação de recorrência, com as condições iniciais, fornece uma definição recursiva da seqüência. Qualquer termo da seqüência pode ser encontrado, a partir das condições iniciais, usando a relação de recorrência um número suficiente de vezes. Entretanto, há maneiras melhores de computar os termos de determinadas classes de seqüências definidas por relações de recorrência e condições iniciais. Discutiremos esses métodos nesta seção e na 7.2.

Modelos com Relações de Recorrência

Podemos usar as relações de recorrência como modelo para uma grande variedade de problemas, como encontrar os juros compostos, contar coelhos em uma ilha, determinar o número de movimentos no quebra-cabeça Torre de Hanói e contar as cadeias de bits com determinadas propriedades.

EXEMPLO 3 **Juros Compostos** Suponha que uma pessoa deposite R$ 10.000,00 em uma poupança com uma taxa de 11% ao ano, com taxa de juros compostos anual. Que valor terá na poupança depois de 30 anos?

Exemplos Extras

Solução: Para resolver esse problema, considere P_n como a quantia na poupança depois de n anos. Como a quantia na poupança depois de n anos é igual à quantia na poupança depois de $n-1$ anos mais os juros do n-ésimo ano, podemos ver que a seqüência $\{P_n\}$ satisfaz a relação de recorrência

$$P_n = P_{n-1} + 0{,}11P_{n-1} = (1{,}11)P_{n-1}.$$

A condição inicial é $P_0 = 10.000$.

Podemos usar uma aproximação repetida para encontrar a fórmula para P_n. Note que

$$\begin{aligned} P_1 &= (1{,}11)P_0 \\ P_2 &= (1{,}11)P_1 = (1{,}11)^2P_0 \\ P_3 &= (1{,}11)P_2 = (1{,}11)^3P_0 \\ &\vdots \\ P_n &= (1{,}11)P_{n-1} = (1{,}11)^n P_0. \end{aligned}$$

Quando inserimos a condição inicial $P_0 = 10.000$, é obtida a fórmula $P_n = (1{,}11)^n 10.000$.

Podemos usar a indução matemática para estabelecer sua validade. Que a fórmula é válida para $n = 0$ é uma conseqüência da condição inicial. Agora, assuma que $P_n = (1{,}11)^n 10.000$. Então, a partir da relação de recorrência e da hipótese indutiva,

$$P_{n+1} = (1{,}11)P_n = (1{,}11)(1{,}11)^n 10.000 = (1{,}11)^{n+1} 10.000.$$

Isto mostra que a fórmula explícita para P_n é válida.

Inserir $n = 30$ na fórmula $P_n = (1{,}11)^n 10.000$ mostra que depois de 30 anos, a poupança terá

$$P_{30} = (1{,}11)^{30} 10.000 = \text{R\$ } 228.922{,}97. \quad \blacktriangleleft$$

O Exemplo 4 mostra como a população de coelhos em uma ilha pode ser modelada usando uma relação de recorrência.

EXEMPLO 4 **Coelhos e os Números de Fibonacci** Considere este problema, proposto originalmente por Leonardo Pisano, também conhecido como Fibonacci, no século XIII, em seu livro *Liber abaci*. Um jovem casal de coelhos é colocado em uma ilha. Um casal de coelhos não procria até que eles tenham dois meses de idade. Depois que eles completam dois meses de idade, cada casal de coelhos produz outro casal todo mês, como mostra a Figura 1. Encontre a relação de recorrência para o número de casais de coelhos na ilha depois de n meses, supondo que nenhum coelho morra.

Links

Solução: Utilize f_n para o número de casais de coelhos depois de n meses. Mostraremos que f_n, $n = 1, 2, 3, \ldots$, são os termos da seqüência de Fibonacci.

A população de coelhos pode ser modelada usando uma relação de recorrência. Ao final do primeiro mês, o número de casais de coelhos na ilha é de $f_1 = 1$. Como este casal não procria durante o segundo mês, $f_2 = 1$ também. Para encontrar o número de casais depois de n meses, adicione o número do mês anterior na ilha, f_{n-1}, ao número de pares recém-nascidos, que é igual a f_{n-2}, porque cada par recém-nascido surge a partir de um casal com, pelo menos, dois meses de idade.

Casais Reprodutores (Com dois meses de idade pelo menos)	Casais Jovens (Com menos de dois meses de idade)	Mês	Casais Reprodutores	Casais Jovens	Total de Casais
		1	0	1	1
		2	0	1	1
		3	1	1	2
		4	1	2	3
		5	2	3	5
		6	3	5	8

FIGURA 1 **Coelhos na Ilha.**

Conseqüentemente, a seqüência $\{f_n\}$ satisfaz a relação de recorrência

$$f_n = f_{n-1} + f_{n-2}$$

para $n \geq 3$ junto com as condições iniciais $f_1 = 1$ e $f_2 = 1$. Como essa relação de recorrência e as condições iniciais determinam esta seqüência, que é única, o número de casais de coelhos na ilha depois de n meses é dado pelo n-ésimo número de Fibonacci. ◀

Demo

O Exemplo 5 envolve um famoso quebra-cabeça.

EXEMPLO 5 **A Torre de Hanói** Um quebra-cabeça popular do final do século XIX, inventado pelo matemático francês Édouard Lucas, chamado de Torre de Hanói, consiste em três pinos colocados sobre um tabuleiro junto com discos de tamanhos diferentes. Inicialmente, esses discos estão colocados no primeiro pino em ordem crescente de tamanho, de cima para baixo (como mostra a Figura 2). As regras do quebra-cabeça permitem que os discos sejam movidos um de cada vez a partir do primeiro pino para outro, contanto que nenhum disco seja colocado acima de um

Links

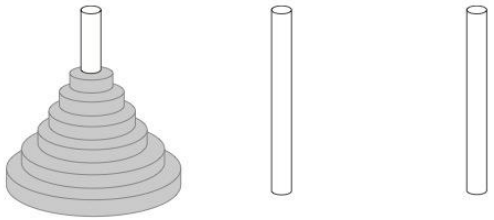


FIGURA 2 **Posição Inicial na Torre de Hanói.**

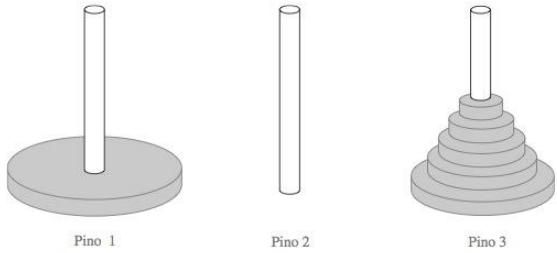


FIGURA 3 Uma Posição Intermediária na Torre de Hanói.

disco menor. O objetivo deste jogo é ter todos os discos no segundo pino em ordem crescente de tamanho.

Considere H_n como o número de movimentos necessários para resolver o quebra-cabeça Torre de Hanói com n discos. Crie uma relação de recorrência para a seqüência $\{H_n\}$.

Solução: Comece com n discos no pino 1. Podemos transferir $n-1$ discos, seguindo as regras do jogo, para o pino 3 usando H_{n-1} movimentos (veja a Figura 3 que mostra os pinos e os discos nesta fase). Mantemos o maior disco fixo durante esses movimentos. Então, usamos um movimento para transferir o maior disco para o segundo pino. Podemos transferir os $n-1$ discos do pino 3 para o pino 2 usando H_{n-1} movimentos adicionais, colocando-os em cima do disco maior, o qual sempre se mantém fixo no fundo do pino 2. Além disso, é fácil ver que o problema não pode ser resolvido usando um número menor de passos. Isto mostra que

$$H_n = 2H_{n-1} + 1.$$

A condição inicial é $H_1 = 1$, porque um disco pode ser transferido do pino 1 para o pino 2, de acordo com as regras do jogo, com um movimento.

Podemos usar um método iterativo para resolver esta relação de recorrência. Note que

$$\begin{aligned} H_n &= 2H_{n-1} + 1 \\ &= 2(2H_{n-2} + 1) + 1 = 2^2H_{n-2} + 2 + 1 \\ &= 2^2(2H_{n-3} + 1) + 2 + 1 = 2^3H_{n-3} + 2^2 + 2 + 1 \\ &\vdots \\ &= 2^{n-1}H_1 + 2^{n-2} + 2^{n-3} + \cdots + 2 + 1 \\ &= 2^{n-1} + 2^{n-2} + \cdots + 2 + 1 \\ &= 2^n - 1. \end{aligned}$$

Usamos a relação de recorrência repetidamente para expressar H_n a partir dos termos anteriores da seqüência. Na penúltima equação, a condição inicial $H_1 = 1$ foi usada. A última equação baseia-se na fórmula para a soma dos termos em uma série geométrica, que pode ser encontrada no Teorema 1 da Seção 2.4.

O método iterativo produziu a solução para a relação de recorrência $H_n = 2H_{n-1} + 1$ com a condição inicial $H_1 = 1$. Esta fórmula pode ser demonstrada usando a indução matemática. Isso foi deixado para o leitor como exercício no final da seção.

A lenda criada para acompanhar o quebra-cabeça diz que, em uma torre em Hanói, macacos transferem 64 discos de ouro do primeiro pino para o seguinte, de acordo com as regras do jogo. A lenda conta que o mundo acabará quando eles terminarem o quebra-cabeça. Quanto tempo

depois que os macacos começaram, o mundo acabará, se os macacos demoram um segundo para movimentar um disco?

A partir da fórmula explícita, os macacos necessitam de

$$2^{64} - 1 = 18.446.744.073.709.551.615$$

movimentos para transferir os discos. Executando um movimento por segundo, os macacos vão demorar mais de 500 bilhões de anos para completar a transferência, assim o mundo sobreviverá ainda um pouco mais. ◀

Lembre-se: Muitas pessoas têm estudado variações do quebra-cabeça original Torre de Hanói, discutido no Exemplo 5. Algumas variantes usam mais pinos, outras permitem que os discos sejam do mesmo tamanho e outras ainda restringem os tipos permitidos de movimento de discos. Uma das variações mais antigas e interessantes é o **Reve's puzzle*,** proposto em 1907 por Henry Dudeney, em seu livro *The Canterbury Puzzles*. O *Reve's puzzle* envolve peregrinos desafiados pelo Governador a mover uma pilha de queijos de diversos tamanhos a partir do primeiro de quatro bancos para outro banco sem colocar um queijo em cima de outro de menor diâmetro. O *Reve's puzzle*, expresso em termos de pinos e discos, segue as mesmas regras do quebra-cabeça Torre de Hanói, exceto pelas quatro pilhas usadas. Você pode achar isso surpreendente, mas ninguém foi capaz de estabelecer um número mínimo de movimentos necessários para resolver este problema para n discos. Entretanto, há uma conjectura, agora com mais de 50 anos, de que o número mínimo de movimentos necessários é igual ao número de movimentos usados por um algoritmo criado por Frame e Stewart em 1939. (Veja os exercícios 54 a 61 no final desta seção e também [St94] para mais informações.)

O Exemplo 6 ilustra como as relações de recorrência podem ser usadas para contar cadeias de bits de diferentes extensões que têm uma determinada propriedade.

EXEMPLO 6 Encontre uma relação de recorrência e dê as condições iniciais para o número de cadeias de bits de extensão n que não têm dois 0s consecutivos. Quantas dessas cadeias têm extensão cinco?

Solução: Considere a_n como o número de cadeias de bits de extensão n que não têm dois zeros consecutivos. Para obter uma relação de recorrência para $\{a_n\}$, note que, pela regra da soma, o número de cadeias de bits de extensão n que não têm dois zeros consecutivos é igual ao número dessas cadeias que terminam em 0 mais o número de cadeias que terminam em 1. Assumiremos que $n \geq 3$, para que a cadeia tenha pelo menos três bits.

As cadeias de bits de extensão n que terminam em 1 e que não têm dois 0s consecutivos são, precisamente, as cadeias de bits de extensão $n - 1$ sem nenhum 0 consecutivo e com um 1 adicionado ao final. Conseqüentemente, há a_{n-1} cadeias desse tipo.

As cadeias de bits de extensão n que terminam em 0 e que não têm dois 0s consecutivos devem ter 1 como seu $(n - 1)$-ésimo bit; se não, elas poderiam terminar em um par de 0s. Temos, então, que as cadeias de bits de extensão n que terminam em 0 e que não têm dois 0s consecutivos são precisamente as cadeias de bits de extensão $n - 2$ sem nenhum 0 consecutivo e com o par 10 adicionado ao final. Conseqüentemente, há a_{n-2} cadeias de bits desse tipo.

Concluímos, como mostra a Figura 4, que

$$a_n = a_{n-1} + a_{n-2}$$

para $n \geq 3$.

As condições iniciais são $a_1 = 2$, porque ambas as cadeias de bits de extensão um, 0 e 1, não têm 0s consecutivos, e $a_2 = 3$, porque as cadeias de bits válidas de extensão dois são 01, 10 e 11.

* *Reve* normalmente é escrito como *reeve*, que é uma palavra arcaica para *governor* – governador.

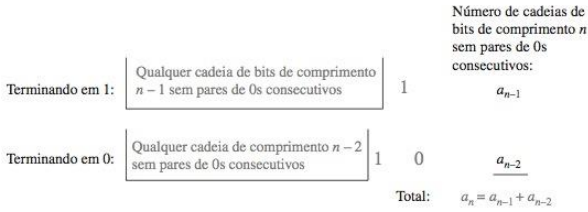


FIGURA 4 Contagem de Cadeias de Bits de Extensão *n* sem Dois 0s Consecutivos.

Para obter a_5, usamos a relação de recorrência três vezes para encontrar

$$a_3 = a_2 + a_1 = 3 + 2 = 5,$$
$$a_4 = a_3 + a_2 = 5 + 3 = 8,$$
$$a_5 = a_4 + a_3 = 8 + 5 = 13.$$ ◀

Lembre-se: $\{a_n\}$ satisfaz a mesma relação de recorrência da seqüência de Fibonacci. Como $a_1 = f_3$ e $a_2 = f_4$, temos que $a_n = f_{n+2}$.

O Exemplo 7 mostra como uma relação de recorrência pode ser utilizada para modelar o número de códigos que estão disponíveis usando determinados registros de validade.

EXEMPLO 7 Enumeração de Códigos Um sistema computacional considera uma cadeia de dígitos decimais como um código válido se ela contém um número par de dígitos 0. Por exemplo, 1230407869 é válido, enquanto 120987045608 não é válido. Considere a_n como o número de códigos de n dígitos válidos. Encontre uma relação de recorrência para a_n.

Solução: Note que $a_1 = 9$ porque há 10 cadeias de um dígito, e apenas uma, a cadeia 0, não é válida. Uma relação de recorrência pode ser derivada para esta seqüência considerando como uma cadeia válida de n dígitos pode ser obtida a partir de $n - 1$ dígitos. Há duas maneiras de formar uma cadeia válida com n dígitos a partir de uma cadeia com um dígito a menos.

Primeiro, uma cadeia válida com n dígitos pode ser obtida adicionando uma cadeia válida de $n - 1$ dígitos com um dígito diferente de 0. Essa adição pode ser feita de nove maneiras. Assim, uma cadeia válida com n dígitos pode ser formada de $9a_{n-1}$ maneiras.

Em segundo lugar, uma cadeia válida de n dígitos pode ser obtida adicionando um 0 à cadeia de extensão $n - 1$, que não é válida. (Isso produz uma cadeia com um número par de dígitos 0, porque a cadeia inválida de extensão $n - 1$ tem um número ímpar de dígitos 0.) O número de maneiras como isto pode ser feito é igual ao número de cadeias inválidas de $(n - 1)$ dígitos. Como há 10^{n-1} cadeias de extensão $n - 1$ e as a_{n-1} são válidas, há $10^{n-1} - a_{n-1}$ cadeias de n dígitos válidas obtidas acrescentando um 0 a uma cadeia inválida de extensão $n - 1$.

Como todas as cadeias válidas de extensão n são produzidas a partir dessas duas maneiras, temos que há

$$\begin{aligned} a_n &= 9a_{n-1} + (10^{n-1} - a_{n-1}) \\ &= 8a_{n-1} + 10^{n-1} \end{aligned}$$

cadeias válidas de extensão n. ◀

O Exemplo 8 estabelece uma relação de recorrência que aparece em diversos contextos.

EXEMPLO 8 Encontre uma relação de recorrência para C_n, o número de maneiras de colocar entre parênteses o produto de $n + 1$ números, $x_0 \cdot x_1 \cdot x_2 \cdot \cdots \cdot x_n$, para especificar a ordem de multiplicação. Por exemplo, $C_3 = 5$, pois há cinco maneiras de colocar entre parênteses $x_0 \cdot x_1 \cdot x_2 \cdot x_3$ para determinar a ordem de multiplicação:

$$((x_0 \cdot x_1) \cdot x_2) \cdot x_3 \qquad (x_0 \cdot (x_1 \cdot x_2)) \cdot x_3 \qquad (x_0 \cdot x_1) \cdot (x_2 \cdot x_3)$$
$$x_0 \cdot ((x_1 \cdot x_2) \cdot x_3) \qquad x_0 \cdot (x_1 \cdot (x_2 \cdot x_3)).$$

Solução: Para desenvolver uma relação de recorrência para C_n, notamos que sempre que inserimos os parênteses no produto $x_0 \cdot x_1 \cdot x_2 \cdot \cdots \cdot x_n$, um operador "$\cdot$" permanece fora de todos os parênteses, ou seja, o operador para a realização da multiplicação final. [Por exemplo, em $(x_0 \cdot (x_1 \cdot x_2)) \cdot x_3$, é o último "$\cdot$" que permanece fora dos parênteses, enquanto em $(x_0 \cdot x_1) \cdot (x_2 \cdot x_3)$ é o segundo "$\cdot$".] Este operador final aparece entre dois desses $n + 1$ números, digamos, x_k e x_{k+1}. Há $C_k C_{n-k-1}$ maneiras de inserir os parênteses para determinar a ordem dos $n + 1$ números para serem multiplicados quando o operador final aparecer entre x_k e x_{k+1}, pois há C_k maneiras de inserir parênteses no produto $x_0 \cdot x_1 \cdot \cdots \cdot x_k$ para determinar quais desses $k + 1$ números serão multiplicados e C_{n-k-1} maneiras de inserir os parênteses no produto $x_{k+1} \cdot x_{k+2} \cdot \cdots \cdot x_n$ para determinar a ordem na qual esses $n - k$ números serão multiplicados. Como este operador final pode aparecer entre quaisquer dois desses $n + 1$ números, temos que

$$\begin{aligned} C_n &= C_0 C_{n-1} + C_1 C_{n-2} + \cdots + C_{n-2} C_1 + C_{n-1} C_0 \\ &= \sum_{k=0}^{n-1} C_k C_{n-k-1}. \end{aligned}$$

Note que as condições iniciais são $C_0 = 1$ e $C_1 = 1$. Esta relação de recorrência pode ser resolvida usando o método de construção de funções, que será discutido na Seção 7.4. Podemos mostrar que $C_n = C(2n, n)/(n + 1)$. (Veja o Exercício 41 no final daquela seção.) ◀

Links A seqüência $\{C_n\}$ é a seqüência de **números de Catalan.** Essa seqüência aparece como solução de muitos problemas diferentes de contagem além deste aqui considerado (para obter detalhes veja o capítulo sobre números de Catalan em [MiRo91] ou em [Ro84a]).

Exercícios

1. Encontre os primeiros cinco termos da seqüência definida pelas relações de recorrência e pelas condições iniciais abaixo.
   a) $a_n = 6a_{n-1}$, $a_0 = 2$
   b) $a_n = a_{n-1}^2$, $a_1 = 2$
   c) $a_n = a_{n-1} + 3a_{n-2}$, $a_0 = 1$, $a_1 = 2$
   d) $a_n = na_{n-1} + n^2 a_{n-2}$, $a_0 = 1$, $a_1 = 1$
   e) $a_n = a_{n-1} + a_{n-3}$, $a_0 = 1$, $a_1 = 2$, $a_2 = 0$
2. Encontre os primeiros seis termos da seqüência definida pelas relações de recorrência e pelas condições iniciais abaixo.
   a) $a_n = -2a_{n-1}$, $a_0 = -1$
   b) $a_n = a_{n-1} - a_{n-2}$, $a_0 = 2$, $a_1 = -1$
   c) $a_n = 3a_{n-1}^2$, $a_0 = 1$
   d) $a_n = na_{n-1} + a_{n-2}^2$, $a_0 = -1$, $a_1 = 0$
   e) $a_n = a_{n-1} - a_{n-2} + a_{n-3}$, $a_0 = 1$, $a_1 = 1$, $a_2 = 2$
3. Considere $a_n = 2^n + 5 \cdot 3^n$ para $n = 0, 1, 2, \ldots$.
   a) Encontre a_0, a_1, a_2, a_3 e a_4.
   b) Mostre que $a_2 = 5a_1 - 6a_0$, $a_3 = 5a_2 - 6a_1$ e $a_4 = 5a_3 - 6a_2$.
   c) Mostre que $a_n = 5a_{n-1} - 6a_{n-2}$ para todos os números inteiros n com $n \geq 2$.
4. Mostre que a seqüência $\{a_n\}$ é uma solução da relação de recorrência $a_n = -3a_{n-1} + 4a_{n-2}$, se
   a) $a_n = 0$.
   b) $a_n = 1$.
   c) $a_n = (-4)^n$.
   d) $a_n = 2(-4)^n + 3$.
5. A seqüência $\{a_n\}$ é uma solução da relação de recorrência $a_n = 8a_{n-1} - 16a_{n-2}$, se
   a) $a_n = 0$?
   b) $a_n = 1$?
   c) $a_n = 2^n$?
   d) $a_n = 4^n$?
   e) $a_n = n4^n$?
   f) $a_n = 2 \cdot 4^n + 3n4^n$?
   g) $a_n = (-4)^n$?
   h) $a_n = n^2 4^n$?

6. Para cada uma das seqüências abaixo, encontre uma relação de recorrência que satisfaça a respectiva seqüência. (As respostas não são únicas, pois há um número infinito de relações de recorrência que satisfazem qualquer seqüência.)

a) $a_n = 3$ **b)** $a_n = 2n$
c) $a_n = 2n + 3$ **d)** $a_n = 5^n$
e) $a_n = n^2$ **f)** $a_n = n^2 + n$
g) $a_n = n + (-1)^n$ **h)** $a_n = n!$

7. Mostre que a seqüência $\{a_n\}$ é uma solução da relação de recorrência $a_n = a_{n-1} + 2a_{n-2} + 2n - 9$, se

a) $a_n = -n + 2$.
b) $a_n = 5(-1)^n - n + 2$.
c) $a_n = 3(-1)^n + 2^n - n + 2$.
d) $a_n = 7 \cdot 2^n - n + 2$.

8. Encontre a solução para cada uma das relações de recorrência abaixo com as condições iniciais dadas. Use uma aproximação iterativa, como a exibida no Exemplo 5.

a) $a_n = -a_{n-1}$, $a_0 = 5$
b) $a_n = a_{n-1} + 3$, $a_0 = 1$
c) $a_n = a_{n-1} - n$, $a_0 = 4$
d) $a_n = 2a_{n-1} - 3$, $a_0 = -1$
e) $a_n = (n + 1)a_{n-1}$, $a_0 = 2$
f) $a_n = 2na_{n-1}$, $a_0 = 3$
g) $a_n = -a_{n-1} + n - 1$, $a_0 = 7$

9. Encontre a solução para cada uma das relações de recorrência e as condições iniciais abaixo. Use um método iterativo, como o apresentado no Exemplo 5.

a) $a_n = 3a_{n-1}$, $a_0 = 2$
b) $a_n = a_{n-1} + 2$, $a_0 = 3$
c) $a_n = a_{n-1} + n$, $a_0 = 1$
d) $a_n = a_{n-1} + 2n + 3$, $a_0 = 4$
e) $a_n = 2a_{n-1} - 1$, $a_0 = 1$
f) $a_n = 3a_{n-1} + 1$, $a_0 = 1$
g) $a_n = na_{n-1}$, $a_0 = 5$
h) $a_n = 2na_{n-1}$, $a_0 = 1$

10. Uma pessoa deposita R$ 1.000,00 em uma poupança que rende 9% de juros compostos por ano.

a) Encontre uma relação de recorrência para a quantia na poupança ao final de n anos.
b) Encontre uma fórmula explícita para a quantia na poupança ao final de n anos.
c) Quanto haverá na poupança depois de 100 anos?

11. Suponha que o número de bactérias em uma colônia triplique a cada hora.

a) Encontre uma relação de recorrência para o número de bactérias depois que se passaram n horas.
b) Se 100 bactérias são usadas para iniciar uma nova colônia, quantas bactérias haverá na colônia em 10 horas?

12. Suponha que a população mundial em 2002 era de 6,2 bilhões e cresce com uma taxa de 1,3% ao ano.

a) Encontre uma relação de recorrência para a população mundial n anos depois de 2002.
b) Encontre uma fórmula explícita para a população mundial n anos depois de 2002.
c) Qual a estimativa da população mundial em 2022?

13. Uma indústria automobilística fabrica carros com uma taxa crescente. No primeiro mês, apenas um carro é fabricado; no segundo mês, dois carros são fabricados, e assim por diante, com n carros fabricados no n-ésimo mês.

a) Encontre uma relação de recorrência para o número de carros produzidos nos primeiros n meses nessa fábrica.
b) Quantos carros são produzidos no primeiro ano?
c) Encontre uma fórmula explícita para o número de carros produzidos nos primeiros n meses por essa fábrica.

14. Um empregado é contratado por uma companhia em 1999 com um salário anual inicial de R$ 50.000,00. Todos os anos esse empregado recebe um aumento de R$ 1.000,00 mais 5% do salário do ano anterior.

a) Encontre uma relação de recorrência para o salário desse empregado n anos depois de 1999.
b) Qual o salário desse empregado em 2007?
c) Encontre uma fórmula explícita para o salário desse empregado n anos depois de 1999.

15. Encontre uma relação de recorrência para o balanço $B(k)$ ao final de k meses de um empréstimo de R$ 5.000,00 com uma taxa de 7%, se for feito um pagamento de R$ 100,00 todo mês. [*Dica:* Expresse $B(k)$ em termos de $B(k - 1)$; os juros mensais são de $(0{,}07/12)B(k - 1)$.]

16. **a)** Encontre uma relação de recorrência para o balanço $B(k)$ ao final de k meses de um empréstimo com taxa de r se for feito um pagamento P todo mês. [*Dica:* Expresse $B(k)$ em termos de $B(k - 1)$ e note que a taxa de juros mensal é de $r/12$.]

b) Determine qual pagamento mensal P deve ser feito para que o empréstimo seja quitado em T meses.

17. Use a indução matemática para verificar a fórmula derivada no Exemplo 5 para o número de movimentos necessários para completar a Torre de Hanói.

18. **a)** Encontre uma relação de recorrência para o número de permutações de um conjunto com n elementos.
b) Use essa relação de recorrência para encontrar o número de permutações de um conjunto com n elementos usando a iteração.

19. Uma máquina de vender livros aceita apenas moedas de um real e notas de R$ 1,00 e R$ 5,00.

a) Encontre uma relação de recorrência para o número de maneiras de depositar n reais na máquina, sendo relevante a ordem na qual as moedas e as cédulas são depositadas.
b) Quais são as condições iniciais?
c) De quantas maneiras é possível depositar R$ 10,00 para comprar um livro?

20. Um país usa como moedas correntes os valores de 1 peso, 2 pesos, 5 pesos e 10 pesos e notas de 5 pesos, 10 pesos, 20 pesos, 50 pesos e 100 pesos. Encontre uma relação de recorrência para o número de maneiras de pagar uma conta de n pesos, sendo relevante a ordem em que as moedas e as cédulas são postas.

21. De quantas maneiras é possível pagar uma conta de 17 pesos usando as moedas correntes descritas no Exercício 20, sendo relevante a ordem em que as moedas e as cédulas são postas?

*22. a) Encontre uma relação de recorrência para o número de seqüências estritamente crescentes de números inteiros positivos que têm o número 1 como seu primeiro termo e n como seu último termo, em que n é um número inteiro positivo, ou seja, seqüências $a_1, a_2, \ldots, a_k$, em que $a_1 = 1$, $a_k = n$ e $a_j < a_{j+1}$ para $j = 1, 2, \ldots, k-1$.
b) Quais são as condições iniciais?
c) Quantas seqüências do tipo descrito em (a) são possíveis quando n é um número inteiro positivo com $n \geq 2$?

23. a) Encontre uma relação de recorrência para o número de cadeias de bits de extensão n que contenham um par de 0s consecutivos.
b) Quais são as condições iniciais?
c) Quantas cadeias de bits de extensão sete contêm dois 0s consecutivos?

24. a) Encontre uma relação de recorrência para o número de cadeias de bits de extensão n que contenham três 0s consecutivos.
b) Quais são as condições iniciais?
c) Quantas cadeias de bits de extensão sete contêm três 0s consecutivos?

25. a) Encontre uma relação de recorrência para o número de cadeias de bits de extensão n que não contenham três 0s consecutivos.
b) Quais são as condições iniciais?
c) Quantas cadeias de bits de extensão sete não contêm três 0s consecutivos?

*26. a) Encontre uma relação de recorrência para o número de cadeias de bits que contenham a seqüência 01.
b) Quais são as condições iniciais?
c) Quantas cadeias de bits de extensão sete contêm a seqüência 01?

27. a) Encontre uma relação de recorrência para o número de maneiras de subir n degraus se a pessoa que estiver subindo as escadas pode subir um ou dois degraus por vez.
b) Quais são as condições iniciais?
c) De quantas maneiras essa pessoa pode subir em um avião com oito degraus?

28. a) Encontre uma relação de recorrência para o número de maneiras de subir n degraus se a pessoa que estiver subindo as escadas pode subir um, dois ou três degraus por vez.
b) Quais são as condições iniciais?
c) De quantas maneiras essa pessoa pode subir em um avião com oito degraus?

Uma cadeia que contém apenas 0s, 1s e 2s é chamada de **cadeia ternária.**

29. a) Encontre uma relação de recorrência para o número de cadeias ternárias que não contenham dois 0s consecutivos.
b) Quais são as condições iniciais?
c) Quantas cadeias ternárias de extensão seis não contêm dois 0s consecutivos?

30. a) Encontre uma relação de recorrência para o número de cadeias ternárias que contenham dois 0s consecutivos.
b) Quais são as condições iniciais?
c) Quantas cadeias ternárias de extensão seis contêm dois 0s consecutivos?

*31. a) Encontre uma relação de recorrência para o número de cadeias ternárias que não contenham dois 0s consecutivos ou dois 1s consecutivos.
b) Quais são as condições iniciais?
c) Quantas cadeias ternárias de extensão seis não contêm dois 0s consecutivos ou dois 1s consecutivos?

*32. a) Encontre uma relação de recorrência para o número de cadeias ternárias que contenham ou dois 0s consecutivos ou dois 1s consecutivos.
b) Quais são as condições iniciais?
c) Quantas cadeias ternárias de extensão seis contêm dois 0s consecutivos ou dois 1s consecutivos?

*33. a) Encontre uma relação de recorrência para o número de cadeias ternárias que não contenham símbolos iguais consecutivos.
b) Quais são as condições iniciais?
c) Quantas cadeias ternárias de extensão seis não contêm símbolos iguais consecutivos?

**34. a) Encontre uma relação de recorrência para o número de cadeias ternárias que contenham dois símbolos iguais consecutivos.
b) Quais são as condições iniciais?
c) Quantas cadeias ternárias de extensão seis contêm símbolos iguais consecutivos?

35. Mensagens são transmitidas por meio de um canal de comunicação que usa dois sinais. A transmissão de um sinal requer 1 microssegundo e a transmissão do outro sinal requer 2 microssegundos.
a) Encontre uma relação de recorrência para o número de mensagens diferentes que consistem em seqüências desses dois sinais, em que cada sinal na mensagem é seguido imediatamente pelo próximo sinal, que pode ser enviado em n microssegundos.
b) Quais são as condições iniciais?
c) Quantas mensagens diferentes podem ser enviadas em 10 microssegundos usando esses dois sinais?

36. Um motorista de ônibus paga todos os pedágios apenas com moedas de 5 e 10 centavos, inserindo uma moeda de cada vez no coletor automático.
a) Encontre uma relação de recorrência para o número de maneiras diferentes de o motorista poder pagar um pedágio de n centavos (em que a ordem na qual cada moeda é usada é relevante).
b) De quantas maneiras diferentes o motorista pode pagar um pedágio de 45 centavos?

37. a) Encontre uma relação de recorrência que satisfaça R_n, em que R_n é o número de regiões em que um plano é dividido em n linhas, se duas linhas não podem ser paralelas e três dessas linhas não podem passar pelo mesmo ponto.
b) Encontre R_n usando a iteração.

*38. a) Encontre uma relação de recorrência que satisfaça R_n, em que R_n é o número de regiões em que uma superfície de uma esfera é dividida por n círculos perfeitos (que são intersecções da esfera e de planos que passam pelo centro da esfera), se três dos círculos perfeitos não passam pelo mesmo ponto.
b) Encontre R_n usando a iteração.

*39. a) Encontre uma relação de recorrência que satisfaça S_n, em que S_n é o número de regiões em que o espaço tridimensional é dividido por n planos, se sempre três des-

ses planos encontram-se em um mesmo ponto, mas quatro desses planos não passam pelo mesmo ponto.

b) Encontre S_n usando a iteração.

40. Encontre uma relação de recorrência para o número de cadeias de bits de extensão n com um número par de 0s.

41. Quantas cadeias de bits de extensão sete contêm um número par de 0s?

42. a) Encontre uma relação de recorrência para o número de maneiras de cobrir completamente um tabuleiro de damas de $2 \times n$ com dominós 1×2. [*Dica:* Considere separadamente a cobertura quando a posição no canto superior direito do tabuleiro estiver coberta com um dominó posicionado horizontalmente e quando estiver coberta com um dominó posicionado verticalmente.]

b) Quais são as condições iniciais para a relação de recorrência do item (a)?

c) De quantas maneiras é possível cobrir completamente um tabuleiro de damas de 2×17 com dominós 1×2?

43. a) Encontre uma relação de recorrência para o número de maneiras de cobrir uma calçada com ladrilhos de ardósia, se os ladrilhos são vermelhos, verdes ou cinza, a fim de que dois ladrilhos vermelhos não sejam adjacentes e que ladrilhos da mesma cor sejam considerados indistinguíveis.

b) Quais são as condições iniciais para a relação de recorrência do item (a)?

c) De quantas maneiras é possível cobrir uma calçada com sete ladrilhos como descrito no item (a)?

44. Mostre que os números de Fibonacci satisfazem a relação de recorrência $f_n = 5f_{n-4} + 3f_{n-5}$ para $n = 5, 6, 7, \ldots$, junto com as condições iniciais $f_0 = 0, f_1 = 1, f_2 = 1, f_3 = 2$ e $f_4 = 3$. Use essa relação de recorrência para mostrar que f_{5n} é divisível por 5, para $n = 1, 2, 3, \ldots$.

***45.** Considere $S(m,n)$ como o número de funções sobrejetivas de um conjunto com m elementos em um conjunto com n elementos. Mostre que $S(m,n)$ satisfaz a relação de recorrência

$$S(m,n) = n^m - \sum_{k=1}^{n-1} C(n,k)S(m,k)$$

sempre que $m \geq n$ e $n > 1$, com a condição inicial $S(m, 1) = 1$.

46. a) Descreva todas as maneiras nas quais os parênteses podem ser colocados no produto $x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4$ para determinar a ordem de multiplicação.

b) Use a relação de recorrência desenvolvida no Exemplo 8 para calcular C_4, o número de maneiras de colocar parênteses no produto de cinco números para determinar a ordem de multiplicação. Verifique se você listou o número correto de maneiras no item (a).

c) Confira seu resultado do item (b) encontrando C_4, usando a fórmula fechada para C_n mencionada na solução do Exemplo 8.

47. a) Use a relação de recorrência desenvolvida no Exemplo 8 para determinar C_5, o número de maneiras de colocar parênteses no produto de seis números para determinar a ordem de multiplicação.

b) Confira seu resultado com a fórmula fechada para C_5 mencionada na solução do Exemplo 8.

***48.** No quebra-cabeça Torre de Hanói, suponha que nosso objetivo seja transferir todos os n discos do pino 1 para o pino 3, mas sem mover um disco diretamente do pino 1 para o 3. Cada movimento deve envolver o pino 2. Como sempre, não podemos colocar um disco maior em cima de um menor.

a) Encontre uma relação de recorrência para o número de movimentos necessários para resolver o problema para n discos com essa restrição adicionada.

b) Resolva esta relação de recorrência a fim de encontrar uma fórmula para o número de movimentos necessários para resolver o problema para n discos.

c) Quantos arranjos são possíveis dos n discos nos três pinos para que nenhum disco maior fique em cima de um menor?

d) Mostre que todo arranjo permitido dos n discos aparece na solução desta variante do problema.

Os exercícios 49 a 53 tratam de uma variação do **problema de Josefo**, descrito por Graham, Knuth e Patashnik em [GrKnPa94]. Esse problema baseia-se no registro do historiador Flavio Josefo, que fazia parte de um grupo de 14 judeus rebeldes que foram pegos em uma caverna por uma armadilha dos romanos durante a Guerra entre Judeus e Romanos no século I. Os rebeldes preferiram suicidar-se a ser capturados; eles decidiram formar um círculo e eliminar as pessoas iterativamente ao longo do círculo, matando sempre o terceiro rebelde vivo à esquerda. Entretanto, Josefo e outro rebelde não queriam ser mortos dessa maneira; eles determinaram as posições onde deveriam permanecer para serem as últimas duas pessoas vivas. A variação que consideramos começa com n pessoas, numeradas de 1 a n, mantidas em um círculo. Em cada etapa, a segunda pessoa à esquerda que estiver viva é eliminada até que reste uma. Indicamos o número de sobreviventes por $J(n)$.

49. Determine o valor de $J(n)$ para cada número inteiro n com $1 \leq n \leq 16$.

50. Use os valores que você encontrou no Exercício 49 para presumir uma fórmula para $J(n)$. [*Dica:* Escreva $n = 2^m + k$, em que m é um número inteiro não negativo e k é um número inteiro não negativo menor que 2^m.]

51. Mostre que $J(n)$ satisfaz a relação de recorrência $J(2n) = 2J(n) - 1$ e $J(2n + 1) = 2J(n) + 1$, para $n \geq 1$ e $J(1) = 1$.

52. Use a indução matemática para demonstrar a fórmula que você presumiu no Exercício 50, usando a relação de recorrência do Exercício 51.

53. Determine $J(100)$, $J(1.000)$ e $J(10.000)$ a partir da sua fórmula para $J(n)$.

Os exercícios 54 a 61 envolvem o *Reve's puzzle*, a variação do quebra-cabeça Torre de Hanói com quatro pinos e n discos. Antes de apresentar os exercícios, descreveremos o algoritmo de Frame–Stewart para movimentar os discos do pino 1 para o pino 4 a fim de que nenhum disco maior seja colocado em cima de um menor. Esse algoritmo, dado como entrada o número de discos n, depende da escolha de um número inteiro k, com $1 \leq k \leq n$. Quando há apenas um disco, movimenta-se ele do pino 1 para o 4 e pára. Para $n > 1$, o algoritmo procede recursi-

vamente, usando três passos. Movimenta-se recursivamente a pilha dos $n-k$ discos menores do pino 1 para o 2, usando-se todos os quatro pinos. Então, movimenta-se a pilha dos k discos maiores do pino 1 para o 4, usando-se o algoritmo dos três pinos da Torre de Hanói sem usar o pino dos $n-k$ discos menores. Por fim, movimenta-se recursivamente os menores $n-k$ discos para o pino 4, usando-se os quatro pinos. Frame e Stewart mostraram que para produzir o menor número de movimentos usando o seu algoritmo, k deve ser o menor número inteiro, tal que n não exceda $t_k = k(k+1)/2$, o k-ésimo número triangular, ou seja, $t_{k-1} < n \leq t_k$. A hipótese modificada, conhecida como **conjectura de Frame,** é a de que esse algoritmo usa o menor número de movimentos necessários para resolver o problema, não importando como os discos são movimentados.

54. Mostre que o *Reve's puzzle* com três discos pode ser resolvido usando cinco, e não menos, movimentos.

55. Mostre que o *Reve's puzzle* com quatro discos pode ser resolvido usando nove, e não menos, movimentos.

56. Descreva os movimentos feitos pelo algoritmo de Frame–Stewart, com k escolhido para que seja usado o menor número de movimentos necessários, para

a) 5 discos.

b) 6 discos.

c) 7 discos.

d) 8 discos.

***57.** Mostre que, se $R(n)$ for o número de movimentos usados pelo algoritmo de Frame–Stewart para resolver o *Reve's puzzle* com n discos, em que k é o menor número inteiro escolhido com $n \leq k(k+1)/2$, então $R(n)$ satisfaz a relação de recorrência $R(n) = 2R(n-k) + 2^k - 1$, com $R(0) = 0$ e $R(1) = 1$.

***58.** Mostre que se k for escolhido sob as condições do Exercício 57, então $R(n) - R(n-1) = 2^{k-1}$.

***59.** Mostre que se k for escolhido sob as condições do Exercício 57, então $R(n) = \sum_{i=1}^{k} i2^{i-1} - (t_k - n)2^{k-1}$.

***60.** Use o Exercício 59 para dar um limite superior no número de movimentos necessários para resolver o *Reve's puzzle* para todos os números inteiros n com $1 \leq n \leq 25$.

***61.** Mostre que $R(n)$ é $O(\sqrt{n}2^{\sqrt{2n}})$.

Considere $\{a_n\}$ como uma seqüência de números reais. As **diferenças atrasadas** dessa seqüência são definidas recursivamente como mostradas a seguir. A **primeira diferença** ∇a_n é

$$\nabla a_n = a_n - a_{n-1}.$$

A **($k+1$)-ésima diferença** $\nabla^{k+1}a_n$ é obtida a partir de $\nabla^k a_n$ por

$$\nabla^{k+1}a_n = \nabla^k a_n - \nabla^k a_{n-1}.$$

62. Encontre ∇a_n para a seqüência $\{a_n\}$, em que

a) $a_n = 4$. **b)** $a_n = 2n$.

c) $a_n = n^2$. **d)** $a_n = 2^n$.

63. Encontre $\nabla^2 a_n$ para a seqüência no Exercício 62.

64. Mostre que $a_{n-1} = a_n - \nabla a_n$.

65. Mostre que $a_{n-2} = a_n - 2\nabla a_n + \nabla^2 a_n$.

***66.** Demonstre que a_{n-k} pode ser expresso em termos de a_n, ∇a_n, $\nabla^2 a_n, \ldots, \nabla^k a_n$.

67. Expresse a relação de recorrência $a_n = a_{n-1} + a_{n-2}$ em termos de a_n, ∇a_n e $\nabla^2 a_n$.

68. Mostre que qualquer relação de recorrência para a seqüência $\{a_n\}$ pode ser escrita em termos de a_n, ∇a_n, $\nabla^2 a_n, \ldots$. A equação resultante que envolve as seqüências e suas diferenças é chamada de **equação de diferença.**

7.2 Resolvendo Relações de Recorrência Lineares

Introdução

Links

Uma grande variedade de relações de recorrência ocorre em modelos. Algumas dessas relações podem ser resolvidas usando iteração ou outra técnica direta. Entretanto, uma importante classe de relações de recorrência pode ser resolvida explicitamente de maneira sistemática. Estas são as relações de recorrência que expressam os termos de uma seqüência como combinações lineares de termos anteriores.

DEFINIÇÃO 1

Uma *relação de recorrência linear e homogênea de grau k com coeficientes constantes* é uma relação de recorrência na forma

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k},$$

em que $c_1, c_2, \ldots, c_k$ são números reais e $c_k \neq 0$.

A relação de recorrência na definição é **linear**, pois o lado direito é a soma dos termos anteriores da seqüência multiplicados por uma função de n. A relação de recorrência é **homogênea**, pois

nenhum termo aparece sem estar multiplicado por a_j. Os coeficientes dos termos da seqüência são todos **constantes,** em vez de funções que dependem de n. O **grau** é k, pois a_n é expresso em termos dos k termos anteriores da seqüência.

Como conseqüência do segundo princípio da indução matemática, temos que uma seqüência, ao satisfazer a relação de recorrência na definição, é determinada unicamente por essa relação de recorrência e pelas k condições iniciais

$$a_0 = C_0, a_1 = C_1, \ldots, a_{k-1} = C_{k-1}.$$

EXEMPLO 1 A relação de recorrência $P_n = (1{,}11)P_{n-1}$ é uma relação de recorrência linear e homogênea de grau um. A relação de recorrência $f_n = f_{n-1} + f_{n-2}$ é uma relação de recorrência linear e homogênea de grau dois. A relação de recorrência $a_n = a_{n-5}$ é uma relação de recorrência linear e homogênea de grau cinco. ◀

O Exemplo 2 apresenta algumas relações de recorrência que não são relações de recorrência lineares e homogêneas com coeficientes constantes.

EXEMPLO 2 A relação de recorrência $a_n = a_{n-1} + a_{n-2}^2$ não é linear. A relação de recorrência $H_n = 2H_{n-1} + 1$ não é homogênea. A relação de recorrência $B_n = nB_{n-1}$ não tem coeficientes constantes. ◀

Relações de recorrência lineares e homogêneas são estudadas por dois motivos. Primeiro, elas geralmente aparecem em modelos para resolução de problemas; segundo, elas podem ser resolvidas sistematicamente, isto é, há um método para resolvê-las.

Resolvendo Relações de Recorrência Lineares e Homogêneas com Coeficientes Constantes

O método básico para resolver relações de recorrência lineares e homogêneas é procurar por soluções na forma $a_n = r^n$, em que r é uma constante. Note que $a_n = r^n$ é uma solução para a relação de recorrência $a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k}$ se e somente se

$$r^n = c_1r^{n-1} + c_2r^{n-2} + \cdots + c_kr^{n-k}.$$

Quando ambos os lados dessa equação são divididos por r^{n-k} e o lado direito é subtraído do lado esquerdo, obtemos a equação

$$r^k - c_1r^{k-1} - c_2r^{k-2} - \cdots - c_{k-1}r - c_k = 0.$$

Conseqüentemente, a seqüência $\{a_n\}$ com $a_n = r^n$ é uma solução se e somente se r for uma solução desta última equação, a qual chamamos de **equação característica** da relação de recorrência. As soluções dessa equação são chamadas de **raízes características** da relação de recorrência. Como veremos, essas raízes características podem ser usadas para fornecer uma fórmula explícita para todas as soluções da relação de recorrência.

Desenvolveremos primeiramente resultados que envolvem relações de recorrência lineares e homogêneas com coeficientes constantes de grau dois. Então, serão verificados resultados gerais nos quais o grau pode ser maior que dois. Como as demonstrações necessárias para estabelecer os resultados nos casos gerais são mais complicadas, elas não serão trabalhadas neste texto.

Voltaremos agora nossa atenção para as relações de recorrência lineares e homogêneas de grau dois. Primeiro, considere o caso em que existem duas raízes características distintas.

TEOREMA 1 Considere c_1 e c_2 como números reais. Suponha que $r^2 - c_1 r - c_2 = 0$ tenha duas raízes distintas r_1 e r_2. Então, a seqüência $\{a_n\}$ é uma solução para a relação de recorrência $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ se e somente se $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ para $n = 0, 1, 2, \ldots$, em que α_1 e α_2 são constantes.

Demonstração: Devemos completar duas etapas para demonstrar este teorema. Primeiro, devemos demonstrar que se r_1 e r_2 forem as raízes da equação característica e α_1 e α_2 forem constantes, então a seqüência $\{a_n\}$ com $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ é uma solução da relação de recorrência. Em segundo lugar, devemos demonstrar que se a seqüência $\{a_n\}$ for uma solução, então $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ para constantes como α_1 e α_2.

Agora, mostraremos que se $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$, então a seqüência $\{a_n\}$ é uma solução da relação de recorrência. Como r_1 e r_2 são raízes de $r^2 - c_1 r - c_2 = 0$, temos que $r_1^2 = c_1 r_1 + c_2$, $r_2^2 = c_1 r_2 + c_2$.

A partir dessas equações, vemos que

$$\begin{aligned} c_1 a_{n-1} + c_2 a_{n-2} &= c_1(\alpha_1 r_1^{n-1} + \alpha_2 r_2^{n-1}) + c_2(\alpha_1 r_1^{n-2} + \alpha_2 r_2^{n-2}) \\ &= \alpha_1 r_1^{n-2}(c_1 r_1 + c_2) + \alpha_2 r_2^{n-2}(c_1 r_2 + c_2) \\ &= \alpha_1 r_1^{n-2} r_1^2 + \alpha_2 r_2^{n-2} r_2^2 \\ &= \alpha_1 r_1^n + \alpha_2 r_2^n \\ &= a_n. \end{aligned}$$

Isto nos demonstra que a seqüência $\{a_n\}$, com $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$, é uma solução da relação de recorrência.

Para demonstrar que toda solução $\{a_n\}$ da relação de recorrência $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ tem $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ para $n = 0, 1, 2, \ldots$, para algum par de constantes α_1 e α_2, suponha que $\{a_n\}$ seja uma solução para a relação de recorrência e as condições iniciais $a_0 = C_0$ e $a_1 = C_1$ sejam mantidas. Mostraremos que há constantes α_1 e α_2, tal que a seqüência $\{a_n\}$, com $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$, satisfaz essas mesmas condições iniciais. Isso requer que

$$\begin{aligned} a_0 &= C_0 = \alpha_1 + \alpha_2, \\ a_1 &= C_1 = \alpha_1 r_1 + \alpha_2 r_2. \end{aligned}$$

Podemos resolver essas duas equações para α_1 e α_2. A partir da primeira equação, temos que $\alpha_2 = C_0 - \alpha_1$. Inserindo essa expressão na segunda equação, temos

$$C_1 = \alpha_1 r_1 + (C_0 - \alpha_1) r_2.$$

Assim,

$$C_1 = \alpha_1(r_1 - r_2) + C_0 r_2.$$

Isto nos mostra que

$$\alpha_1 = \frac{C_1 - C_0 r_2}{r_1 - r_2}$$

e

$$\alpha_2 = C_0 - \alpha_1 = C_0 - \frac{C_1 - C_0 r_2}{r_1 - r_2} = \frac{C_0 r_1 - C_1}{r_1 - r_2},$$

em que essas expressões para α_1 e α_2 dependem do fato de que $r_1 \neq r_2$. (Quando $r_1 = r_2$, esse teorema não é verdadeiro.) Assim, com esses valores para α_1 e α_2, a seqüência $\{a_n\}$ com $\alpha_1 r_1^n + \alpha_2 r_2^n$ satisfaz as duas condições iniciais.

Sabemos que $\{a_n\}$ e $\{\alpha_1 r_1^n + \alpha_2 r_2^n\}$ são, ambas, soluções para a relação de recorrência $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ e satisfazem as condições iniciais quando $n = 0$ e $n = 1$. Como há uma única solução para a relação de recorrência linear e homogênea de grau dois com duas condições iniciais, temos que as duas soluções são idênticas, ou seja, $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$, para todo número inteiro n não negativo. Completamos, dessa maneira, a demonstração evidenciando que a solução da relação de recorrência linear e homogênea com coeficientes constantes de grau dois deve ter a forma $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$, com α_1 e α_2 constantes. ◁

As raízes características de uma relação de recorrência linear e homogênea com coeficientes constantes podem ser números complexos. O Teorema 1 ainda se aplica a este caso (o que também ocorre com os teoremas subseqüentes desta seção). As relações de recorrência com raízes características complexas não serão discutidas neste texto. Os leitores familiarizados com os números complexos podem desejar resolver os exercícios 38 e 39 no final desta seção.

Os exemplos 3 e 4 mostram como utilizar o Teorema 1 para resolver as relações de recorrência.

EXEMPLO 3 Qual é a solução da relação de recorrência

$$a_n = a_{n-1} + 2a_{n-2}$$

com $a_0 = 2$ e $a_1 = 7$?

Exemplos Extras

Solução: O Teorema 1 pode ser utilizado para resolver este problema. A equação característica da relação de recorrência é $r^2 - r - 2 = 0$. Suas raízes são $r = 2$ e $r = -1$. Assim, a seqüência $\{a_n\}$ é uma solução da relação de recorrência se e somente se

$$a_n = \alpha_1 2^n + \alpha_2 (-1)^n,$$

para constantes como α_1 e α_2. A partir das condições iniciais, temos que

$$a_0 = 2 = \alpha_1 + \alpha_2,$$
$$a_1 = 7 = \alpha_1 \cdot 2 + \alpha_2 \cdot (-1).$$

Resolvendo essas duas equações, temos que $\alpha_1 = 3$ e $\alpha_2 = -1$. Assim, a solução para a relação de recorrência e condições iniciais é a seqüência $\{a_n\}$, com

$$a_n = 3 \cdot 2^n - (-1)^n.$$ ◀

EXEMPLO 4 Encontre uma fórmula explícita para a seqüência de Fibonacci.

Solução: Lembre-se que a seqüência de números de Fibonacci satisfaz a relação de recorrência $f_n = f_{n-1} + f_{n-2}$, bem como as condições iniciais $f_0 = 0$ e $f_1 = 1$. As raízes da equação característica $r^2 - r - 1 = 0$ são $r_1 = (1 + \sqrt{5})/2$ e $r_2 = (1 - \sqrt{5})/2$. Portanto, a partir do Teorema 1, temos que os números de Fibonacci são dados por

$$f_n = \alpha_1 \left(\frac{1 + \sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1 - \sqrt{5}}{2}\right)^n,$$

para constantes como α_1 e α_2. As condições iniciais $f_0 = 0$ e $f_1 = 1$ podem ser utilizadas para encontrar essas constantes. Temos

$$f_0 = \alpha_1 + \alpha_2 = 0,$$
$$f_1 = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right) = 1.$$

A solução dessas equações, simultaneamente para α_1 e α_2, é

$$\alpha_1 = 1/\sqrt{5}, \qquad \alpha_2 = -1/\sqrt{5}.$$

Conseqüentemente, os números de Fibonacci são dados por

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n.$$

◀

O Teorema 1 não se aplica quando há uma raiz característica de multiplicidade dois. Se isso ocorrer, então $a_n = nr_0^n$ é a segunda solução para a relação de recorrência quando r_0 for uma raiz de multiplicidade dois da equação característica. O Teorema 2 mostra como trabalhar com este caso.

TEOREMA 2

Considere c_1 e c_2 como números reais com $c_2 \neq 0$. Suponha que $r^2 - c_1 r - c_2 = 0$ tenha apenas uma raiz r_0. Uma seqüência $\{a_n\}$ é uma solução da relação de recorrência $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ se e somente se $a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$, para $n = 0, 1, 2, \ldots$, em que α_1 e α_2 são constantes.

A demonstração do Teorema 2 foi deixada como exercício no final desta seção. O Exemplo 5 ilustra o uso desse teorema.

EXEMPLO 5 Qual é a solução da relação de recorrência

$$a_n = 6a_{n-1} - 9a_{n-2}$$

com condições iniciais $a_0 = 1$ e $a_1 = 6$?

Solução: A única raiz de $r^2 - 6r + 9 = 0$ é $r = 3$. Assim, a solução para essa relação de recorrência é

$$a_n = \alpha_1 3^n + \alpha_2 n 3^n$$

para constantes como α_1 e α_2. Usando as condições iniciais, temos que

$$a_0 = 1 = \alpha_1,$$
$$a_1 = 6 = \alpha_1 \cdot 3 + \alpha_2 \cdot 3.$$

Resolvendo essas duas equações, temos que $\alpha_1 = 1$ e $\alpha_2 = 1$. Conseqüentemente, a solução para essa relação de recorrência com essas condições iniciais é

$$a_n = 3^n + n3^n.$$

◀

Vamos agora expressar o resultado universal sobre a solução das relações de recorrência lineares e homogêneas com coeficientes constantes, nas quais o grau pode ser maior que dois,

considerando a hipótese de que a equação característica tenha raízes distintas. A demonstração desse resultado foi proposta como exercício para o leitor.

TEOREMA 3 Considere $c_1, c_2, \ldots, c_k$ como números reais. Suponha que a equação característica

$$r^k - c_1 r^{k-1} - \cdots - c_k = 0$$

tenha k raízes distintas $r_1, r_2, \ldots, r_k$. Então, a seqüência $\{a_n\}$ é uma solução para a relação de recorrência

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$$

se e somente se

$$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \cdots + \alpha_k r_k^n$$

para $n = 0, 1, 2, \ldots$, em que $\alpha_1, \alpha_2, \ldots, \alpha_k$ são constantes.

Ilustramos o uso do teorema acima com o Exemplo 6.

EXEMPLO 6 Encontre a solução para a relação de recorrência

$$a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$$

com as condições iniciais $a_0 = 2$, $a_1 = 5$ e $a_2 = 15$.

Solução: O polinômio característico dessa relação de recorrência é

$$r^3 - 6r^2 + 11r - 6.$$

As raízes características são $r = 1$, $r = 2$ e $r = 3$, pois $r^3 - 6r^2 + 11r - 6 = (r - 1)(r - 2)(r - 3)$. Assim, as soluções para essa relação de recorrência encontram-se na forma

$$a_n = \alpha_1 \cdot 1^n + \alpha_2 \cdot 2^n + \alpha_3 \cdot 3^n.$$

Para encontrar as constantes α_1, α_2 e α_3, utilizamos as condições iniciais. Estas nos fornecem

$$\begin{aligned} a_0 &= 2 = \alpha_1 + \alpha_2 + \alpha_3, \\ a_1 &= 5 = \alpha_1 + \alpha_2 \cdot 2 + \alpha_3 \cdot 3, \\ a_2 &= 15 = \alpha_1 + \alpha_2 \cdot 4 + \alpha_3 \cdot 9. \end{aligned}$$

Quando este sistema de equações é resolvido em α_1, α_2 e α_3, encontramos $\alpha_1 = 1$, $\alpha_2 = -1$ e $\alpha_3 = 2$. Assim, a única solução para essa relação de recorrência com essas condições iniciais é a seqüência $\{a_n\}$, com

$$a_n = 1 - 2^n + 2 \cdot 3^n. \quad \blacktriangleleft$$

Expressaremos agora o resultado mais geral sobre relações de recorrência lineares e homogêneas com coeficientes constantes, que permitem que a equação característica tenha raízes múltiplas. O ponto principal é que para cada raiz r da equação característica, a solução geral

tem uma soma da forma $P(n)r^n$, em que $P(n)$ é um polinômio de grau $m - 1$, sendo m a multiplicidade dessa raiz. Deixaremos a demonstração desse resultado como um exercício desafiador para o leitor.

TEOREMA 4 Considere $c_1, c_2, \ldots, c_k$ como números reais. Suponha que a equação característica

$$r^k - c_1 r^{k-1} - \cdots - c_k = 0$$

tenha t raízes distintas $r_1, r_2, \ldots, r_t$ com multiplicidades $m_1, m_2, \ldots, m_t$, respectivamente, sendo $m_i \geq 1$ para $i = 1, 2, \ldots, t$ e $m_1 + m_2 + \cdots + m_t = k$. Então, a seqüência $\{a_n\}$ é uma solução para a relação de recorrência

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$$

se e somente se

$$\begin{aligned} a_n = {} & (\alpha_{1,0} + \alpha_{1,1} n + \cdots + \alpha_{1,m_1-1} n^{m_1-1}) r_1^n \\ & + (\alpha_{2,0} + \alpha_{2,1} n + \cdots + \alpha_{2,m_2-1} n^{m_2-1}) r_2^n \\ & + \cdots + (\alpha_{t,0} + \alpha_{t,1} n + \cdots + \alpha_{t,m_t-1} n^{m_t-1}) r_t^n \end{aligned}$$

para $n = 0, 1, 2, \ldots$, em que $\alpha_{i,j}$ são constantes para $1 \leq i \leq t$ e $0 \leq j \leq m_i - 1$.

O Exemplo 7 ilustra como o Teorema 4 é utilizado para encontrar a forma generalizada de solução de uma relação de recorrência linear e homogênea quando a equação característica tem diversas raízes repetidas.

EXEMPLO 7 Suponha que as raízes da equação característica de uma relação de recorrência linear e homogênea sejam 2, 2, 2, 5, 5 e 9 (ou seja, há três raízes, a raiz 2 de multiplicidade três, a raiz 5 de multiplicidade dois e a raiz 9 de multiplicidade um). Qual é a forma da solução geral?

Solução: A partir do Teorema 4, a forma geral da solução é

$$(\alpha_{1,0} + \alpha_{1,1} n + \alpha_{1,2} n^2) 2^n + (\alpha_{2,0} + \alpha_{2,1} n) 5^n + \alpha_{3,0} 9^n. \quad \blacktriangleleft$$

Ilustraremos agora o uso do Teorema 4 na resolução de uma relação de recorrência linear e homogênea com coeficientes constantes quando a equação característica tem uma raiz de multiplicidade três.

EXEMPLO 8 Encontre a solução para a relação de recorrência

$$a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$$

com condições iniciais $a_0 = 1$, $a_1 = -2$ e $a_2 = -1$.

Solução: A equação característica dessa relação de recorrência é

$$r^3 + 3r^2 + 3r + 1 = 0.$$

Como $r^3 + 3r^2 + 3r + 1 = (r + 1)^3$, há uma única raiz $r = -1$ de multiplicidade três na equação característica. A partir do Teorema 4, as soluções dessa relação de recorrência encontram-se na forma

$$a_n = \alpha_{1,0}(-1)^n + \alpha_{1,1}n(-1)^n + \alpha_{1,2}n^2(-1)^n.$$

Para encontrar as constantes $\alpha_{1,0}$, $\alpha_{1,1}$ e $\alpha_{1,2}$, usamos as condições iniciais. Ou seja,

$$a_0 = 1 = \alpha_{1,0},$$
$$a_1 = -2 = -\alpha_{1,0} - \alpha_{1,1} - \alpha_{1,2},$$
$$a_2 = -1 = \alpha_{1,0} + 2\alpha_{1,1} + 4\alpha_{1,2}.$$

A solução desse sistema de equações é $\alpha_{1,0} = 1$, $\alpha_{1,1} = 3$ e $\alpha_{1,2} = -2$. Assim, a solução única para essa relação de recorrência e condições iniciais é a seqüência $\{a_n\}$, com

$$a_n = (1 + 3n - 2n^2)(-1)^n.$$ ◀

Relações de Recorrência Lineares e Heterogêneas com Coeficientes Constantes

Vimos até agora como resolver relações de recorrência lineares e homogêneas com coeficientes constantes. Contudo, queremos saber se há uma técnica relativamente simples para resolver uma relação de recorrência linear, mas não homogênea, com coeficientes constantes, tal como $a_n = 3a_{n-1} + 2n$. Veremos que a resposta é afirmativa para determinadas famílias de relações de recorrência.

A relação de recorrência $a_n = 3a_{n-1} + 2n$ é um exemplo de **relação de recorrência linear e heterogênea com coeficientes constantes,** ou seja, uma relação de recorrência cuja forma é

$$a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k} + F(n),$$

em que $c_1, c_2, \ldots, c_k$ são números reais e $F(n)$ é uma função não nula dependente apenas de n. A relação de recorrência

$$a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k}$$

é chamada de **relação de recorrência homogênea associada.** Ela desenvolve um importante papel na solução da relação de recorrência heterogênea.

EXEMPLO 9 Cada relação de recorrência $a_n = a_{n-1} + 2^n$, $a_n = a_{n-1} + a_{n-2} + n^2 + n + 1$, $a_n = 3a_{n-1} + n3^n$ e $a_n = a_{n-1} + a_{n-2} + a_{n-3} + n!$ é linear e heterogênea com coeficientes constantes. As relações de recorrência associadas lineares e homogêneas são $a_n = a_{n-1}$, $a_n = a_{n-1} + a_{n-2}$, $a_n = 3a_{n-1}$ e $a_n = a_{n-1} + a_{n-2} + a_{n-3}$, respectivamente. ◀

A questão principal das relações de recorrência lineares e heterogêneas com coeficientes constantes é que toda solução é a soma de uma solução particular com uma solução da relação de recorrência associada linear e homogênea, como mostra o Teorema 5.

TEOREMA 5 Se $\{a_n^{(p)}\}$ é uma solução particular para a relação de recorrência linear e heterogênea com coeficientes constantes

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n),$$

então, toda solução se apresenta sob a forma de $\{a_n^{(p)} + a_n^{(h)}\}$, em que $\{a_n^{(h)}\}$ é uma solução para a relação de recorrência associada homogênea

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}.$$

Demonstração: Como $\{a_n^{(p)}\}$ é uma solução particular para a relação de recorrência heterogênea, sabemos que

$$a_n^{(p)} = c_1 a_{n-1}^{(p)} + c_2 a_{n-2}^{(p)} + \cdots + c_k a_{n-k}^{(p)} + F(n).$$

Suponha agora que $\{b_n\}$ seja uma segunda solução para a relação de recorrência heterogênea, sendo

$$b_n = c_1 b_{n-1} + c_2 b_{n-2} + \cdots + c_k b_{n-k} + F(n).$$

Subtraindo a primeira equação da segunda, temos que

$$b_n - a_n^{(p)} = c_1(b_{n-1} - a_{n-1}^{(p)}) + c_2(b_{n-2} - a_{n-2}^{(p)}) + \cdots + c_k(b_{n-k} - a_{n-k}^{(p)}).$$

Portanto, $\{b_n - a_n^{(p)}\}$ é uma solução para a relação de recorrência associada linear e homogênea, ou seja, $\{a_n^{(h)}\}$. Conseqüentemente, $b_n = a_n^{(p)} + a_n^{(h)}$ para todo n. ◁

A partir do Teorema 5, vemos que a chave para resolver as relações de recorrência heterogêneas com coeficientes constantes é encontrar uma solução particular. Então, toda solução será a soma desta com uma solução da relação de recorrência associada homogênea. Embora não haja nenhum método que generalize uma solução que seja válida para toda função $F(n)$, existem técnicas que funcionam para determinados tipos de funções $F(n)$, tais como polinômios e potências de constantes. Isso será ilustrado nos exemplos 10 e 11.

EXEMPLO 10 Encontre todas as soluções para a relação de recorrência $a_n = 3a_{n-1} + 2n$. Qual é a solução com $a_1 = 3$?

Solução: Para resolver essa relação de recorrência linear e heterogênea com coeficientes constantes, precisamos encontrar sua relação de recorrência associada linear e homogênea e uma solução particular para essa equação heterogênea. A equação associada linear e homogênea é $a_n = 3a_{n-1}$. Suas soluções são $a_n^{(h)} = \alpha 3^n$, em que α é uma constante.

Encontraremos agora uma solução particular. Como $F(n) = 2n$ é um polinômio de grau um em n, uma solução razoável é uma função linear em n, suponhamos $p_n = cn + d$, em que c e d são constantes. Para determinar se há muitas soluções com esta forma, suponha que $p_n = cn + d$ seja uma solução. Então, a equação $a_n = 3a_{n-1} + 2n$ passa a ser $cn + d = 3(c(n-1) + d) + 2n$. Simplificando e reorganizando os termos dados, temos $(2 + 2c)n + (2d - 3c) = 0$. Temos que $cn + d$ é uma solução se e somente se $2 + 2c = 0$ e $2d - 3c = 0$. Isso nos mostra que $cn + d$ é uma solução se e somente se $c = -1$ e $d = -3/2$. Conseqüentemente, $a_n^{(p)} = -n - 3/2$ é uma solução particular.

A partir do Teorema 5, todas as soluções têm a forma

$$a_n = a_n^{(p)} + a_n^{(h)} = -n - \frac{3}{2} + \alpha \cdot 3^n,$$

em que α é uma constante.

Para encontrar a solução com $a_1 = 3$, considere $n = 1$ na fórmula que obtivemos para a solução geral. Encontramos que $3 = -1 - 3/2 + 3\alpha$, o qual implica $\alpha = 11/6$. A solução que procurávamos é $a_n = -n - 3/2 + (11/6)3^n$. ◀

EXEMPLO 11 Encontre todas as soluções para a relação de recorrência

Exemplos Extras

$$a_n = 5a_{n-1} - 6a_{n-2} + 7^n.$$

Solução: Esta é uma relação de recorrência linear e heterogênea. As soluções de sua relação de recorrência associada homogênea

$$a_n = 5a_{n-1} - 6a_{n-2}$$

são $a_n^{(h)} = \alpha_1 \cdot 3^n + \alpha_2 \cdot 2^n$, em que α_1 e α_2 são constantes. Como $F(n) = 7^n$, uma solução razoável seria $a_n^{(p)} = C \cdot 7^n$, em que C é uma constante. Substituindo os termos desta seqüência na relação de recorrência temos $C \cdot 7^n = 5C \cdot 7^{n-1} - 6C \cdot 7^{n-2} + 7^n$. Simplificando a equação por 7^{n-2}, temos $49C = 35C - 6C + 49$, o que implica $20C = 49$ ou $C = 49/20$.
Assim, $a_n^{(p)} = (49/20)7^n$ é uma solução particular. A partir do Teorema 5, todas as soluções estão na forma

$$a_n = \alpha_1 \cdot 3^n + \alpha_2 \cdot 2^n + (49/20)7^n.$$ ◀

Nos exemplos 10 e 11, propusemos uma hipótese didática de que existiam soluções de uma determinada forma. Em ambos os casos, fomos capazes de encontrar essas soluções. Isto não foi por acaso. Sempre que $F(n)$ for o produto de um polinômio em n e a n-ésima potência de uma constante, sabemos exatamente qual forma a solução terá, como afirmado no Teorema 6. Deixaremos a demonstração do Teorema 6 como um exercício desafiador para o leitor.

TEOREMA 6 Suponha que $\{a_n\}$ satisfaça a relação de recorrência linear e heterogênea

$$a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k} + F(n),$$

em que $c_1, c_2, \ldots, c_k$ são números reais, e

$$F(n) = (b_tn^t + b_{t-1}n^{t-1} + \cdots + b_1n + b_0)s^n,$$

em que $b_0, b_1, \ldots, b_t$ e s são também números reais. Quando s não for uma raiz da equação característica da relação de recorrência associada linear e homogênea, haverá uma solução particular na forma

$$(p_tn^t + p_{t-1}n^{t-1} + \cdots + p_1n + p_0)s^n.$$

Quando s for uma raiz dessa equação característica e sua multiplicidade for m, haverá uma solução particular na forma

$$n^m(p_tn^t + p_{t-1}n^{t-1} + \cdots + p_1n + p_0)s^n.$$

Note que, neste caso, quando s é uma raiz de multiplicidade m para a equação característica da relação de recorrência associada linear e homogênea, o fator n^m assegura que a solução proposta não será uma solução para a relação de recorrência associada linear e homogênea.

A seguir, fornecemos o Exemplo 12 para ilustrar a forma de uma solução particular dada pelo Teorema 6.

EXEMPLO 12 Qual a forma de uma solução particular da relação de recorrência linear e heterogênea $a_n = 6a_{n-1} - 9a_{n-2} + F(n)$, quando $F(n) = 3^n$, $F(n) = n3^n$, $F(n) = n^2 2^n$ e $F(n) = (n^2 + 1)3^n$?

Solução: A relação de recorrência associada linear e homogênea é $a_n = 6a_{n-1} - 9a_{n-2}$. Sua equação característica, $r^2 - 6r + 9 = (r - 3)^2 = 0$, tem apenas uma raiz, 3, de multiplicidade dois. Para aplicar o Teorema 6, com $F(n)$ na forma $P(n)s^n$, em que $P(n)$ é um polinômio e s é uma constante, precisamos avaliar se s é uma raiz dessa equação característica.

Como $s = 3$ é uma raiz com multiplicidade $m = 2$, mas $s = 2$ não é uma raiz, o Teorema 6 nos diz que uma solução particular tem a forma $p_0 n^2 3^n$ se $F(n) = 3^n$; $n^2(p_1 n + p_0)3^n$ se $F(n) = n3^n$; $(p_2 n^2 + p_1 n + p_0)2^n$ se $F(n) = n^2 2^n$ e $n^2(p_2 n^2 + p_1 n + p_0)3^n$ se $F(n) = (n^2 + 1)3^n$. ◀

Devemos ser cuidadosos quando $s = 1$ ao resolver as relações de recorrência que são abrangidas pelo Teorema 6. Em particular, para aplicar esse teorema com $F(n) = b_t n_t + b_{t-1} n_{t-1} + \cdots + b_1 n + b_0$, o parâmetro s assume o valor $s = 1$ (mesmo quando o termo 1^n não aparece explicitamente). A partir do teorema, a forma da solução depende de 1 ser uma raiz da equação característica da relação de recorrência associada linear e homogênea. Isso será ilustrado no Exemplo 13, que mostra como o Teorema 6 pode ser utilizado para encontrar uma fórmula para a soma dos primeiros n números inteiros positivos.

EXEMPLO 13 Considere a_n como a soma dos primeiros n números inteiros positivos, para que

$$a_n = \sum_{k=1}^{n} k.$$

Note que a_n satisfaz a relação de recorrência linear e heterogênea

$$a_n = a_{n-1} + n.$$

(Para obter a_n, a soma dos primeiros n números inteiros positivos, a partir de a_{n-1}, a soma dos primeiros $n-1$ números inteiros positivos, adicionamos n.) Note que a condição inicial é $a_1 = 1$.

A relação de recorrência associada linear e homogênea para a_n é

$$a_n = a_{n-1}.$$

As soluções para essa relação de recorrência homogênea são dadas por $a_n^{(h)} = c(1)^n = c$, em que c é uma constante. Para encontrar todas as soluções para $a_n = a_{n-1} + n$, precisamos encontrar apenas uma única solução. A partir do Teorema 6, como $F(n) = n = n \cdot (1)^n$ e $s = 1$ é uma raiz de grau um da equação característica das relações de recorrência associadas lineares e homogêneas, há uma solução particular na forma de $n(p_1 n + p_0) = p_1 n^2 + p_0 n$.

Inserindo essa relação na relação de recorrência, temos $p_1 n^2 + p_0 n = p_1(n-1)^2 + p_0(n-1) + n$. Simplificando, vemos que $n(2p_1 - 1) + (p_0 - p_1) = 0$, o que significa que $2p_1 - 1 = 0$ e $p_0 - p_1 = 0$, então $p_0 = p_1 = 1/2$. Assim,

$$a_n^{(p)} = \frac{n^2}{2} + \frac{n}{2} = \frac{n(n+1)}{2}$$

é uma solução particular. Portanto, todas as soluções da relação de recorrência original $a_n = a_{n-1} + n$ são dadas por $a_n = a_n^{(h)} + a_n^{(p)} = c + n(n+1)/2$. Como $a_1 = 1$, temos $1 = a_1 = c + 1 \cdot 2/2 = c + 1$, assim, $c = 0$. Temos que $a_n = n(n+1)/2$. (Esta é a mesma fórmula dada na Tabela 2 da Seção 2.4, derivada anteriormente.) ◀

Exercícios

1. Determine quais das relações abaixo são relações de recorrência lineares e homogêneas com coeficientes constantes. Encontre também seus respectivos graus.
- **a)** $a_n = 3a_{n-1} + 4a_{n-2} + 5a_{n-3}$
- **b)** $a_n = 2na_{n-1} + a_{n-2}$
- **c)** $a_n = a_{n-1} + a_{n-4}$
- **d)** $a_n = a_{n-1} + 2$
- **e)** $a_n = a_{n-1}^2 + a_{n-2}$
- **f)** $a_n = a_{n-2}$
- **g)** $a_n = a_{n-1} + n$

2. Determine quais das relações abaixo são relações de recorrência lineares e homogêneas com coeficientes constantes. Encontre também seus respectivos graus.
- **a)** $a_n = 3a_{n-2}$
- **b)** $a_n = 3$
- **c)** $a_n = a_{n-1}^2$
- **d)** $a_n = a_{n-1} + 2a_{n-3}$
- **e)** $a_n = a_{n-1}/n$
- **f)** $a_n = a_{n-1} + a_{n-2} + n + 3$
- **g)** $a_n = 4a_{n-2} + 5a_{n-4} + 9a_{n-7}$

3. Resolva as relações de recorrência abaixo com as condições iniciais dadas.
- **a)** $a_n = 2a_{n-1}$ para $n \geq 1$, $a_0 = 3$
- **b)** $a_n = a_{n-1}$ para $n \geq 1$, $a_0 = 2$
- **c)** $a_n = 5a_{n-1} - 6a_{n-2}$ para $n \geq 2$, $a_0 = 1$, $a_1 = 0$
- **d)** $a_n = 4a_{n-1} - 4a_{n-2}$ para $n \geq 2$, $a_0 = 6$, $a_1 = 8$
- **e)** $a_n = -4a_{n-1} - 4a_{n-2}$ para $n \geq 2$, $a_0 = 0$, $a_1 = 1$
- **f)** $a_n = 4a_{n-2}$ para $n \geq 2$, $a_0 = 0$, $a_1 = 4$
- **g)** $a_n = a_{n-2}/4$ para $n \geq 2$, $a_0 = 1$, $a_1 = 0$

4. Resolva as relações de recorrência abaixo com as condições iniciais dadas.
- **a)** $a_n = a_{n-1} + 6a_{n-2}$ para $n \geq 2$, $a_0 = 3$, $a_1 = 6$
- **b)** $a_n = 7a_{n-1} - 10a_{n-2}$ para $n \geq 2$, $a_0 = 2$, $a_1 = 1$
- **c)** $a_n = 6a_{n-1} - 8a_{n-2}$ para $n \geq 2$, $a_0 = 4$, $a_1 = 10$
- **d)** $a_n = 2a_{n-1} - a_{n-2}$ para $n \geq 2$, $a_0 = 4$, $a_1 = 1$
- **e)** $a_n = a_{n-2}$ para $n \geq 2$, $a_0 = 5$, $a_1 = -1$
- **f)** $a_n = -6a_{n-1} - 9a_{n-2}$ para $n \geq 2$, $a_0 = 3$, $a_1 = -3$
- **g)** $a_{n+2} = -4a_{n+1} + 5a_n$ para $n \geq 0$, $a_0 = 2$, $a_1 = 8$

5. Quantas mensagens diferentes podem ser transmitidas em n microssegundos, usando os dois sinais descritos no Exercício 35 da Seção 7.1?

6. Quantas mensagens diferentes podem ser transmitidas em n microssegundos, usando três sinais diferentes, se um sinal requer 1 microssegundo para a transmissão e os outros dois sinais, 2 microssegundos cada um para a transmissão, e um sinal na mensagem é seguido imediatamente pelo próximo sinal?

7. De quantas maneiras um tabuleiro retangular de damas $2 \times n$ pode ser ladrilhado, usando peças de 1×2 e 2×2?

8. Um modelo para o número de lagostas capturadas por ano baseia-se na hipótese de que o número de lagostas pescadas em um ano é a média do número da pesca dos dois anos anteriores.
- **a)** Encontre uma relação de recorrência para $\{L_n\}$, em que L_n é o número de lagostas capturadas em n anos, seguindo a hipótese para este modelo.
- **b)** Encontre L_n se 100 000 lagostas foram capturadas no ano 1 e 300 000 no ano 2.

9. Um depósito de R$ 100.000,00 é feito em um fundo de investimentos no começo do ano. No último dia do ano, dois dividendos são ganhos. O primeiro dividendo é 20% da quantia no fundo durante aquele ano. O segundo dividendo é 45% da quantia no fundo no ano anterior.
- **a)** Encontre uma relação de recorrência para $\{P_n\}$, em que P_n é a quantia no fundo ao final de n anos, se não houver nenhum saque.
- **b)** Qual é a quantia no fundo depois de n anos se não houver nenhum saque?

***10.** Demonstre o Teorema 2.

11. Os **números de Lucas** satisfazem a relação de recorrência

$$L_n = L_{n-1} + L_{n-2},$$

e as condições iniciais $L_0 = 2$ e $L_1 = 1$.
- **a)** Mostre que $L_n = f_{n-1} + f_{n+1}$ para $n = 2, 3, \ldots$, em que f_n é o n-ésimo número de Fibonacci.
- **b)** Encontre uma fórmula explícita para os números de Lucas.

12. Encontre a solução para $a_n = 2a_{n-1} + a_{n-2} - 2a_{n-3}$ para $n = 3, 4, 5, \ldots$, com $a_0 = 3$, $a_1 = 6$ e $a_2 = 0$.

13. Encontre a solução para $a_n = 7a_{n-2} + 6a_{n-3}$, com $a_0 = 9$, $a_1 = 10$ e $a_2 = 32$.

14. Encontre a solução para $a_n = 5a_{n-2} - 4a_{n-4}$, com $a_0 = 3$, $a_1 = 2$, $a_2 = 6$ e $a_3 = 8$.

15. Encontre a solução para $a_n = 2a_{n-1} + 5a_{n-2} - 6a_{n-3}$, com $a_0 = 7$, $a_1 = -4$ e $a_2 = 8$.

***16.** Demonstre o Teorema 3.

17. Demonstre a identidade abaixo relacionada aos números de Fibonacci e aos coeficientes binomiais:

$$f_{n+1} = C(n, 0) + C(n-1, 1) + \cdots + C(n-k, k),$$

em que n é um número inteiro positivo e $k = \lfloor n/2 \rfloor$. [*Dica:* Considere $a_n = C(n, 0) + C(n-1, 1) + \cdots + C(n-k, k)$. Mostre que a seqüência $\{a_n\}$ satisfaz a mesma relação de recorrência e as condições iniciais da seqüência dos números de Fibonacci.]

18. Resolva a relação de recorrência $a_n = 6a_{n-1} - 12a_{n-2} + 8a_{n-3}$, com $a_0 = -5$, $a_1 = 4$ e $a_2 = 88$.

19. Resolva a relação de recorrência $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$, com $a_0 = 5$, $a_1 = -9$ e $a_2 = 15$.

20. Encontre a forma geral das soluções da relação de recorrência $a_n = 8a_{n-2} - 16a_{n-4}$.

21. Qual é a forma geral das soluções de uma relação de recorrência linear e homogênea, se sua equação característica tem as raízes 1, 1, 1, 1, −2, −2, −2, 3, 3, −4?

22. Qual é a forma geral das soluções de uma relação de recorrência linear e homogênea, se sua equação característica tem as raízes −1, −1, −1, 2, 2, 5, 5, 7?

23. Considere a seguinte relação de recorrência linear e heterogênea $a_n = 3a_{n-1} + 2^n$.

a) Mostre que $a_n = -2^{n+1}$ é uma solução dessa relação de recorrência.

b) Use o Teorema 5 para encontrar todas as soluções dessa relação de recorrência.

c) Encontre a solução para $a_0 = 1$.

24. Considere a seguinte relação de recorrência linear e heterogênea $a_n = 2a_{n-1} + 2^n$.

a) Mostre que $a_n = n2^n$ é uma solução dessa relação de recorrência.

b) Use o Teorema 5 para encontrar todas as soluções dessa relação de recorrência.

c) Encontre a solução para $a_0 = 2$.

25. a) Determine os valores das constantes A e B, tal que $a_n = An + B$ é uma solução da relação de recorrência $a_n = 2a_{n-1} + n + 5$.

b) Use o Teorema 5 para encontrar todas as soluções dessa relação de recorrência.

c) Encontre a solução para essa relação de recorrência para $a_0 = 4$.

26. Qual é a forma geral de uma solução particular garantida pelo Teorema 6 para a relação de recorrência linear e heterogênea $a_n = 6a_{n-1} - 12a_{n-2} + 8a_{n-3} + F(n)$, se

a) $F(n) = n^2$?

b) $F(n) = 2^n$?

c) $F(n) = n2^n$?

d) $F(n) = (-2)^n$?

e) $F(n) = n^2 2^n$?

f) $F(n) = n^3(-2)^n$?

g) $F(n) = 3$?

27. Qual é a forma geral de uma solução particular garantida pelo Teorema 6 para a relação de recorrência linear e heterogênea $a_n = 8a_{n-2} - 16a_{n-4} + F(n)$, se

a) $F(n) = n^3$?

b) $F(n) = (-2)^n$?

c) $F(n) = n2^n$?

d) $F(n) = n^2 4^n$?

e) $F(n) = (n^2 - 2)(-2)^n$?

f) $F(n) = n^4 2^n$?

g) $F(n) = 2$?

28. a) Encontre todas as soluções para a relação de recorrência $a_n = 2a_{n-1} + 2n^2$.

b) Encontre a solução para a relação de recorrência do item (a) com a condição inicial $a_1 = 4$.

29. a) Encontre todas as soluções para a relação de recorrência $a_n = 2a_{n-1} + 3^n$.

b) Encontre a solução para a relação de recorrência do item (a) com a condição inicial $a_1 = 5$.

30. a) Encontre todas as soluções para a relação de recorrência $a_n = -5a_{n-1} - 6a_{n-2} + 42 \cdot 4^n$.

b) Encontre a solução dessa relação de recorrência com $a_1 = 56$ e $a_2 = 278$.

31. Encontre todas as soluções para a relação de recorrência $a_n = 5a_{n-1} - 6a_{n-2} + 2^n + 3n$. [*Dica:* Procure por uma solução particular na forma $qn2^n + p_1 n + p_2$, em que q, p_1 e p_2 são constantes.]

32. Encontre a solução para a relação de recorrência $a_n = 2a_{n-1} + 3 \cdot 2^n$.

33. Encontre todas as soluções para a relação de recorrência $a_n = 4a_{n-1} - 4a_{n-2} + (n+1)2^n$.

34. Encontre todas as soluções para a relação de recorrência $a_n = 7a_{n-1} - 16a_{n-2} + 12a_{n-3} + n4^n$, com $a_0 = -2$, $a_1 = 0$ e $a_2 = 5$.

35. Encontre a solução para a relação de recorrência $a_n = 4a_{n-1} - 3a_{n-2} + 2^n + n + 3$, com $a_0 = 1$ e $a_1 = 4$.

36. Considere a_n como a soma dos primeiros n quadrados perfeitos, ou seja, $a_n = \sum_{k=1}^{n} k^2$. Mostre que a seqüência $\{a_n\}$ satisfaz a relação de recorrência linear e heterogênea $a_n = a_{n-1} + n^2$ e a condição inicial $a_1 = 1$. Use o Teorema 6 para determinar uma fórmula para a_n, resolvendo esta relação de recorrência.

37. Considere a_n como a soma dos primeiros n números triangulares, ou seja, $a_n = \sum_{k=1}^{n} t_k$, em que $t_k = k(k+1)/2$. Mostre que $\{a_n\}$ satisfaz a relação de recorrência linear e heterogênea $a_n = a_{n-1} + n(n+1)/2$ e a condição inicial $a_1 = 1$. Use o Teorema 6 para determinar uma fórmula para a_n, resolvendo esta relação de recorrência.

38. a) Encontre as raízes características da relação de recorrência linear e homogênea $a_n = 2a_{n-1} - 2a_{n-2}$. (*Nota:* Estão incluídos números complexos.)

b) Encontre a solução para a relação de recorrência do item (a) com $a_0 = 1$ e $a_1 = 2$.

***39. a)** Encontre as raízes características da relação de recorrência linear e homogênea $a_n = a_{n-4}$. (*Nota:* Incluem números complexos.)

b) Encontre a solução da relação de recorrência do item (a), com $a_0 = 1$, $a_1 = 0$, $a_2 = -1$ e $a_3 = 1$.

***40.** Resolva as relações de recorrência simultâneas

$$a_n = 3a_{n-1} + 2b_{n-1}$$

$$b_n = a_{n-1} + 2b_{n-1}$$

com $a_0 = 1$ e $b_0 = 2$.

***41. a)** Utilize a fórmula encontrada no Exemplo 4 para f_n, o n-ésimo número de Fibonacci, para mostrar que f_n é o número inteiro mais próximo de

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n.$$

b) Determine para qual n, f_n é maior que

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$$

e para qual n, f_n é menor que

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n.$$

42. Mostre que se $a_n = a_{n-1} + a_{n-2}$, $a_0 = s$ e $a_1 = t$, em que s e t são constantes, então $a_n = s f_{n-1} + t f_n$ para todos os números inteiros positivos n.

43. Expresse a solução da relação de recorrência linear e heterogênea $a_n = a_{n-1} + a_{n-2} + 1$ para $n \geq 2$, em que $a_0 = 0$ e $a_1 = 1$, em termos de números de Fibonacci. [*Dica:* onsidere $b_n = a_n + 1$ e aplique o Exercício 42 para a seqüência b_n.]

***44.** (*Requer álgebra linear*) Considere $\mathbf{A}_n$ como a matriz $n \times n$ com 2s em sua diagonal principal, 1s em todas as posições próximas à diagonal e 0s nos lugares restantes. Encontre uma relação de recorrência para d_n, o determinante de $\mathbf{A}_n$. Resolva esta relação de recorrência para encontrar uma fórmula para d_n.

45. Suponha que cada par de espécies geneticamente modificadas de coelhos deixados em uma ilha reproduz dois novos pares de coelhos no primeiro mês de idade e seis novos pares de coelhos nos próximos meses. Nenhum dos coelhos morre ou foge da ilha.

a) Encontre a relação de recorrência para o número de pares de coelhos na ilha depois de n meses que um casal recém-nascido é deixado na ilha.

b) A partir da resolução da relação de recorrência do item (a), determine o número de pares de coelhos na ilha depois de n meses que um par é deixado na ilha.

46. Suponha que haja inicialmente um casal de bode e cabra em uma ilha. O número de animais dobra todo ano por reprodução natural e alguns animais são colocados ou removidos todo ano.

a) Construa uma relação de recorrência para o número de cabras/bodes na ilha no início do n-ésimo ano, assumindo que, durante cada ano, 100 animais são adicionados à ilha.

b) Resolva a relação de recorrência do item (a) para encontrar o número de cabras/bodes na ilha no início do n-ésimo ano.

c) Construa uma relação de recorrência para o número de cabras/bodes na ilha no início do n-ésimo ano, assumindo que n animais são retirados durante o n-ésimo ano, para todo $n \geq 3$.

d) Resolva a relação de recorrência do item (c) para o número de cabras/bodes na ilha no início do n-ésimo ano.

47. Uma funcionária nova em uma grande indústria de softwares tem um salário inicial de R$ 50.000,00 por ano e é prometido a ela que ao final de cada ano seu salário irá dobrar, com um valor adicional de R$ 10.000,00 para cada ano que ela permanecer na fábrica.

a) Construa uma relação de recorrência para o salário da funcionária no seu n-ésimo ano de emprego.

b) Resolva esta relação de recorrência para encontrar o valor do seu salário ao final do n-ésimo ano de emprego.

Algumas relações de recorrência lineares que não possuem coeficientes constantes podem ser resolvidas de forma sistemática. Esse é o caso das relações de recorrência com a forma $f(n)a_n = g(n)a_{n-1} + h(n)$. Os exercícios 48 a 50 mostram esta questão.

***48.** **a)** Mostre que a relação de recorrência

$$f(n)a_n = g(n)a_{n-1} + h(n),$$

para $n \geq 1$ e com $a_0 = C$, pode ser reduzida à relação de recorrência na forma

$$b_n = b_{n-1} + Q(n)h(n),$$

em que $b_n = g(n+1)Q(n+1)a_n$, com

$$Q(n) = (f(1)f(2)\cdots f(n-1))/(g(1)g(2)\cdots g(n)).$$

b) Utilize o item (a) para resolver a relação de recorrência original e obter

$$a_n = \frac{C + \sum_{i=1}^{n} Q(i)h(i)}{g(n+1)Q(n+1)}.$$

***49.** Utilize o Exercício 48 para resolver a relação de recorrência $(n+1)a_n = (n+3)a_{n-1} + n$, para $n \geq 1$, com $a_0 = 1$.

50. Podemos mostrar que o número médio de comparações feitas pelo algoritmo *quick sort* (descrito no preâmbulo do Exercício 50 da Seção 4.4), ao selecionar n elementos de forma aleatória, satisfaz a relação de recorrência

$$C_n = n + 1 + \frac{2}{n}\sum_{k=0}^{n-1} C_k$$

para $n = 1, 2, \ldots$, com a condição inicial $C_0 = 0$.

a) Mostre que $\{C_n\}$ também satisfaz a relação de recorrência $nC_n = (n+1)C_{n-1} + 2n$ para $n = 1, 2, \ldots$.

b) Utilize o Exercício 48 para resolver a relação de recorrência do item (a) para encontrar uma fórmula explícita para C_n.

****51.** Demonstre o Teorema 4.

****52.** Demonstre o Teorema 6.

53. Resolva a relação de recorrência $T(n) = nT^2(n/2)$ com a condição inicial $T(1) = 6$. [*Dica:* Considere $n = 2^k$ e então substitua $a_k = \log T(2^k)$ para obter uma relação de recorrência linear e heterogênea.]

7.3 Algoritmos de Divisão e Resolução e Relações de Recorrência

Introdução

Links

Muitos algoritmos recursivos trabalham com um problema a partir da entrada dada e dividem-no em um ou mais problemas menores. Essa redução é aplicada sucessivamente até que as soluções dos problemas menores possam ser encontradas rapidamente. Por exemplo, realizamos uma busca binária a partir da redução da busca de um elemento em uma lista para a busca desse elemento em uma lista com metade do tamanho da original. Aplicamos sucessivamente essa redução até que reste um elemento. Quando ordenamos uma lista de números inteiros usando a merge sort, dividimos a lista em duas metades de tamanhos iguais; ordenamos cada metade separadamente e, então, unimos as duas metades ordenadas. Outro exemplo desse tipo de algoritmo recursivo é um procedimento para multiplicar números inteiros que reduz o problema da multiplicação de dois números inteiros em três multiplicações de pares de inteiros com metade da quantidade de bits. Essa redução é aplicada sucessivamente até que inteiros com um bit sejam obtidos. Esses procedimentos são chamados de **algoritmos de divisão e resolução** porque eles *dividem* um problema em uma ou mais instâncias de tamanho menor e *resolvem* o problema usando as soluções dos problemas menores para encontrar a solução do problema original, talvez com algum trabalho adicional.

Nesta seção, mostraremos como as relações de recorrência podem ser usadas para analisar a complexidade computacional dos algoritmos de divisão e resolução. Utilizaremos essas relações de recorrência para estimar o número de operações usadas por diferentes algoritmos de divisão e resolução, incluindo muitos que serão introduzidos nesta seção.

Relações de Recorrência de Divisão e Resolução

Suponha que um algoritmo recursivo divida um problema de tamanho n em a subproblemas, em que cada subproblema seja do tamanho n/b (para simplificar, suponha que n seja um múltiplo de b; na realidade, os problemas menores são normalmente de tamanho próximo a n/b). Suponha também que seja necessário um total de $g(n)$ operações adicionais na etapa de resolução do algoritmo para combinar a solução dos subproblemas com a solução do problema original. Então, se $f(n)$ representa o número de operações necessárias para resolver o problema de tamanho n, temos que f satisfaz a relação de recorrência

$$f(n) = af(n/b) + g(n).$$

Essa é a chamada **relação de recorrência de divisão e resolução.**

Primeiramente prepararemos as relações de recorrência de divisão e resolução que podem ser utilizadas para estudar a complexidade de alguns algoritmos importantes. Então, mostraremos como utilizar essas relações de recorrência para estimar a complexidade dos algoritmos.

EXEMPLO 1 Exemplos Extras

Busca Binária Introduzimos o algoritmo de busca binária na Seção 3.1. Esse algoritmo reduz a busca de um elemento em uma seqüência de busca de tamanho n em uma busca binária desse elemento em uma seqüência de tamanho $n/2$, quando n for par. (Assim, o problema de tamanho n é reduzido em *um* problema de tamanho $n/2$.) Duas comparações são necessárias para implementar essa redução (uma para determinar qual metade da lista será utilizada e outra para determinar se restará algum termo na lista). Assim, se $f(n)$ é o número de comparações necessárias para procurar um elemento em uma seqüência de tamanho n, então

$$f(n) = f(n/2) + 2$$

quando n for par. ◀

EXEMPLO 2 **Encontrando o Máximo e o Mínimo de uma Seqüência** Considere o seguinte algoritmo para localizar os elementos máximo e mínimo de uma seqüência $a_1, a_2, \ldots, a_n$. Se $n = 1$, então a_1 é o máximo e o mínimo. Se $n > 1$, separe a seqüência em duas, na qual ou as duas terão o mesmo número de elementos ou uma das seqüências terá um elemento a mais que a outra. O problema é reduzido para encontrar o máximo e o mínimo de cada uma das seqüências menores. A solução para o problema original resulta da comparação dos máximos e dos mínimos das duas seqüências menores para obter aquela que abrange o máximo e o mínimo.

Considere $f(n)$ como o número total de comparações necessárias para encontrar os elementos máximo e mínimo da seqüência com n elementos. Mostramos que um problema de tamanho n pode ser reduzido em dois problemas de tamanho $n/2$, quando n for par, usando duas comparações, uma para comparar os máximos das duas seqüências e a outra para comparar os mínimos das duas seqüências. Isso nos fornece a relação de recorrência

$$f(n) = 2f(n/2) + 2$$

quando n for par. ◀

EXEMPLO 3 **Merge Sort** O algoritmo de merge sort (introduzido na Seção 4.4) separa uma lista a ser ordenada com n itens, em que n é par, em duas listas com $n/2$ elementos cada e utiliza menos que n comparações para unir as duas listas ordenadas de $n/2$ itens em uma única lista. Conseqüentemente, o número de comparações utilizadas pela merge sort para ordenar uma lista de n elementos é menor que $M(n)$, em que a função $M(n)$ satisfaz a relação de recorrência de divisão e resolução

$$M(n) = 2M(n/2) + n.$$ ◀

EXEMPLO 4 **Multiplicação Rápida de Números Inteiros** Surpreendentemente, há mais algoritmos eficientes do que o algoritmo convencional (descrito na Seção 3.6) para multiplicar números inteiros. Um desses algoritmos, que utiliza uma técnica de divisão e resolução, será descrito aqui. Esse algoritmo de multiplicação rápida funciona a partir da divisão de dois números inteiros com $2n$ bits em dois blocos, cada um com n bits. Então, a multiplicação original é reduzida da multiplicação de dois números inteiros de $2n$ bits em três multiplicações de inteiros com n bits, mais as adições e as substituições.

Suponha que a e b sejam números inteiros com expansões binárias de extensão $2n$ (adicione bits zeros no início dessas expansões, se for necessário, para que elas fiquem com a mesma extensão). Considere

$$a = (a_{2n-1}a_{2n-2}\cdots a_1a_0)_2 \qquad \text{e} \qquad b = (b_{2n-1}b_{2n-2}\cdots b_1b_0)_2.$$

Considere

$$a = 2^n A_1 + A_0, \qquad b = 2^n B_1 + B_0,$$

em que

$$A_1 = (a_{2n-1}\cdots a_{n+1}a_n)_2, \qquad A_0 = (a_{n-1}\cdots a_1a_0)_2,$$
$$B_1 = (b_{2n-1}\cdots b_{n+1}b_n)_2, \qquad B_0 = (b_{n-1}\cdots b_1b_0)_2.$$

O algoritmo para a multiplicação rápida de números inteiros baseia-se no fato de que ab pode ser escrito como

$$ab = (2^{2n} + 2^n)A_1B_1 + 2^n(A_1 - A_0)(B_0 - B_1) + (2^n + 1)A_0B_0.$$

O fato importante sobre esta identidade é que ela mostra que a multiplicação de dois inteiros de $2n$ bits pode ser trabalhada usando três multiplicações de inteiros de n bits, juntamente com adições, subtrações e substituições. Isso mostra que se $f(n)$ for o número total de operações de bits necessárias para multiplicar dois números inteiros de n bits, então

$$f(2n) = 3f(n) + Cn.$$

O raciocínio por trás dessa equação é o seguinte: as três multiplicações de inteiros de n bits são efetuadas usando operações de $3f(n)$ operações binárias. Cada uma das adições, das subtrações e das substituições utiliza um múltiplo constante de operações de n bits e Cn representa o número total de operações binárias usadas por essas operações. ◀

EXEMPLO 5 **Multiplicação Rápida de Matriz** No Exemplo 5 da Seção 3.8, mostramos que multiplicar duas matrizes $n \times n$ usando a definição de multiplicação de matrizes requer n^3 multiplicações e $n^2(n-1)$ adições. Conseqüentemente, computar o produto de duas matrizes $n \times n$ desta maneira requer $O(n^3)$ operações (multiplicações e adições). Surpreendentemente, há algoritmos de divisão e resolução mais eficientes para multiplicar duas matrizes $n \times n$. Esse algoritmo, inventado por V. Strassen em 1969, reduz a multiplicação de duas matrizes $n \times n$, quando n for par, em sete multiplicações de duas matrizes $(n/2) \times (n/2)$ e 15 adições de matrizes $(n/2) \times (n/2)$. (Veja [CoLeRiSt01] para obter mais detalhes sobre este algoritmo.) Assim, se $f(n)$ for o número de operações (multiplicações e adições) utilizadas, temos que

$$f(n) = 7f(n/2) + 15n^2/4$$

quando n for par. ◀

Como mostram os exemplos 1 a 5, as relações de recorrência na forma $f(n) = af(n/b) + g(n)$ aparecem em muitas situações diferentes. É possível derivar estimativas do tamanho das funções que satisfazem essas relações de recorrência. Suponha que f satisfaça esta relação de recorrência sempre que n for divisível por b. Considere $n = b^k$, em que k é um número inteiro positivo. Então,

$$\begin{aligned} f(n) &= af(n/b) + g(n) \\ &= a^2 f(n/b^2) + ag(n/b) + g(n) \\ &= a^3 f(n/b^3) + a^2 g(n/b^2) + ag(n/b) + g(n) \\ &\vdots \\ &= a^k f(n/b^k) + \sum_{j=0}^{k-1} a^j g(n/b^j). \end{aligned}$$

Como $n/b^k = 1$, temos que

$$f(n) = a^k f(1) + \sum_{j=0}^{k-1} a^j g(n/b^j).$$

Podemos utilizar esta equação para $f(n)$ a fim de estimar o tamanho das funções que satisfazem as relações de divisão e resolução.

TEOREMA 1 Considere f como uma função crescente que satisfaça a relação de recorrência

$$f(n) = af(n/b) + c$$

sempre que n é divisível por b, em que $a \geq 1$, b é um número inteiro maior que 1 e c é um número real positivo. Então,

$$f(n) \text{ é } \begin{cases} O(n^{\log_b a}) \text{ se } a > 1, \\ O(\log n) \text{ se } a = 1. \end{cases}$$

Além disso, quando $n = b^k$, em que k é um número inteiro positivo,

$$f(n) = C_1 n^{\log_b a} + C_2,$$

em que $C_1 = f(1) + c/(a-1)$ e $C_2 = -c/(a-1)$.

Demonstração: Primeiro, considere $n = b^k$. A partir da expressão para $f(n)$, obtida na discussão anterior ao teorema, com $g(n) = c$, temos

$$f(n) = a^k f(1) + \sum_{j=0}^{k-1} a^j c = a^k f(1) + c\sum_{j=0}^{k-1} a^j.$$

Considere, primeiro, o caso quando $a = 1$. Então,

$$f(n) = f(1) + ck.$$

Como $n = b^k$, temos $k = \log_b n$. Assim,

$$f(n) = f(1) + c \log_b n.$$

Quando n não for uma potência de b, temos $b^k < n < b^{k+1}$, para um número inteiro positivo k. Como f é crescente, temos que $f(n) \leq f(b^{k+1}) = f(1) + c(k+1) = (f(1) + c) + ck \leq (f(1) + c) + c \log_b n$. Além disso, em ambos os casos, $f(n)$ é $O(\log n)$ quando $a = 1$.

Agora suponha que $a > 1$. Suponha primeiro que $n = b^k$, em que k é um número inteiro positivo. A partir da fórmula da soma dos termos de uma progressão geométrica (Teorema 1 da Seção 2.4), temos que

$$\begin{aligned} f(n) &= a^k f(1) + c(a^k - 1)/(a-1) \\ &= a^k[f(1) + c/(a-1)] - c/(a-1) \\ &= C_1 n^{\log_b a} + C_2, \end{aligned}$$

pois $a^k = a^{\log_b n} = n^{\log_b a}$ (veja o Exercício 4 do Apêndice 2), em que $C_1 = f(1) + c/(a-1)$ e $C_2 = -c/(a-1)$.

Suponha agora que n não seja uma potência de b. Então, $b^k < n < b^{k+1}$, em que k é um número inteiro não negativo. Como f é crescente,

$$\begin{aligned} f(n) &\leq f(b^{k+1}) = C_1 a^{k+1} + C_2 \\ &\leq (C_1 a) a^{\log_b n} + C_2 \\ &\leq (C_1 a) n^{\log_b a} + C_2, \end{aligned}$$

pois $k \leq \log_b n < k + 1$.

Assim, temos que $f(n)$ é $O(n^{\log_b a})$. ◁

Os exemplos 6 a 9 ilustram como o Teorema 1 é utilizado.

EXEMPLO 6 Considere $f(n) = 5f(n/2) + 3$ e $f(1) = 7$. Encontre $f(2^k)$, em que k é um número inteiro positivo. Faça também a estimativa de $f(n)$, se f for uma função crescente.

Exemplos Extras

Solução: A partir da demonstração do Teorema 1, com $a = 5$, $b = 2$ e $c = 3$, vemos que, se $n = 2^k$, então

$$\begin{aligned} f(n) &= a^k [f(1) + c/(a-1)] + [-c/(a-1)] \\ &= 5^k[7 + (3/4)] - 3/4 \\ &= 5^k(31/4) - 3/4. \end{aligned}$$

Da mesma forma, se $f(n)$ for crescente, o Teorema 1 mostra que $f(n)$ é $O(n^{\log_b a}) = O(n^{\log 5})$. ◀

Podemos usar o Teorema 1 para estimar a complexidade computacional do algoritmo de busca binária e do algoritmo dado no Exemplo 2 para localizar o máximo e o mínimo de uma seqüência.

EXEMPLO 7 Faça a estimativa do número de comparações utilizadas por uma busca binária.

Solução: No Exemplo 1, mostramos que $f(n) = f(n/2) + 2$, quando n for par, em que f é o número de comparações necessárias para realizar uma busca binária de uma seqüência de tamanho n. Assim, a partir do Teorema 1, temos que $f(n)$ é $O(\log n)$. ◀

EXEMPLO 8 Faça a estimativa do número de comparações utilizadas para localizar os elementos máximo e mínimo em uma seqüência usando o algoritmo dado no Exemplo 2.

Solução: No Exemplo 2, mostramos que $f(n) = 2f(n/2) + 2$, quando n for par, em que f é o número de comparações necessárias para este algoritmo. Assim, a partir do Teorema 1, temos que $f(n)$ é $O(n^{\log 2}) = O(n)$. ◀

Agora, veremos um teorema mais geral, e mais complicado, do qual o Teorema 1 é um caso especial. Esse teorema (ou suas versões mais poderosas, incluindo a estimativa Theta-grande) é, muitas vezes, conhecido como *Master Theorem*, ou simplesmente teorema principal, pois é bastante útil na análise de complexidade de muitos algoritmos importantes de divisão e de resolução.

TEOREMA 2 **MASTER THEOREM** Considere f como uma função crescente que satisfaz a relação de recorrência

$$f(n) = af(n/b) + cn^d$$

sempre que $n = b^k$, em que k é um número inteiro positivo, $a \geq 1$, b é um número inteiro maior que 1 e c e d são números reais, sendo c positivo e d não negativo. Então,

$$f(n) \text{ é } \begin{cases} O(n^d) & \text{se } a < b^d, \\ O(n^d \log n) & \text{se } a = b^d, \\ O(n^{\log_b a}) & \text{se } a > b^d. \end{cases}$$

A demonstração do Teorema 2 está proposta como exercício (exercícios 29 a 33) para o leitor no final desta seção.

EXEMPLO 9 **Complexidade da Merge Sort** No Exemplo 3, explicamos que o número de comparações utilizado pela merge sort para ordenar uma lista de n elementos é menor que $M(n)$, em que $M(n) = 2M(n/2) + n$. A partir do *Master Theorem* (Teorema 2), encontramos que $M(n)$ é $O(n \log n)$, concordando com a estimativa encontrada na Seção 4.4. ◀

EXEMPLO 10 Faça a estimativa do número de operações de bits necessárias para multiplicar dois números inteiros com n bits usando o algoritmo de multiplicação rápida descrito no Exemplo 4.

Solução: O Exemplo 4 mostra que $f(n) = 3f(n/2) + Cn$, quando n for par, em que $f(n)$ é o número de operações de bits necessárias para multiplicar dois números inteiros com n bits usando o algoritmo de multiplicação rápida. Assim, a partir do *Master Theorem* (Teorema 2), temos que $f(n)$ é $O(n^{\log 3})$. Note que $\log 3 \sim 1{,}6$. Como o algoritmo convencional para a multiplicação usa $O(n^2)$ operações de bits, o algoritmo de multiplicação rápida é uma melhoria substancial no algoritmo convencional em termos da complexidade temporal para números inteiros suficientemente grandes que aparecem em aplicações práticas. ◀

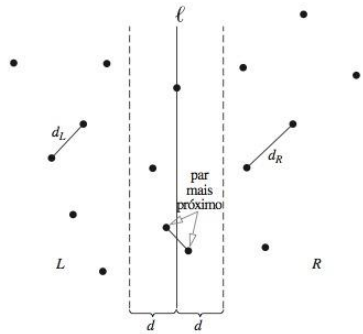
EXEMPLO 11 Faça a estimativa do número de multiplicações e adições necessárias para multiplicar duas matrizes $n \times n$ usando o algoritmo de multiplicação de matrizes ao qual nos referimos no Exemplo 5.

Solução: Considere $f(n)$ como o número de adições e multiplicações usadas pelo algoritmo mencionado no Exemplo 5 para multiplicar duas matrizes $n \times n$. Temos que $f(n) = 7f(n/2) + 15n^2/4$, quando n for par. Assim, a partir do *Master Theorem* (Teorema 2), temos que $f(n)$ é $O(n^{\log 7})$. Note que $\log 7 \sim 2{,}8$. Como o algoritmo convencional para a multiplicação de duas matrizes $n \times n$ usa $O(n^3)$ adições e multiplicações, temos que para números inteiros n suficientemente grandes, incluindo aqueles que aparecem em muitas aplicações práticas, esse algoritmo é substancialmente mais eficiente em complexidade temporal, se comparado com o algoritmo convencional. ◀

O PROBLEMA DO PAR DE PONTOS MAIS PRÓXIMO Concluímos esta seção introduzindo um algoritmo de divisão e resolução da geometria computacional, uma parte da matemática discreta dedicada aos algoritmos que resolvem problemas geométricos.

EXEMPLO 12 **O Problema do Par de Pontos Mais Próximo** Considere o problema de determinar o par de pontos mais próximo em um conjunto de n pontos $(x_1, y_1), \ldots, (x_n, y_n)$ no plano, em que a distância entre dois pontos (x_i, y_i) e (x_j, y_j) é a distância normal euclidiana $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

Links



Nesta ilustração, o problema de encontrar o par mais próximo em um conjunto de 16 pontos é reduzido a dois problemas de encontrar o par mais próximo em cada conjunto de oito pontos e ao problema de determinar se há pontos mais próximos de $d = \text{mind}(d_L, d_R)$ dentro da faixa de largura $2d$ centrado em ℓ.

FIGURA 1 O Passo Recursivo do Algoritmo para Resolução do Problema do Par de Pontos Mais Próximo.

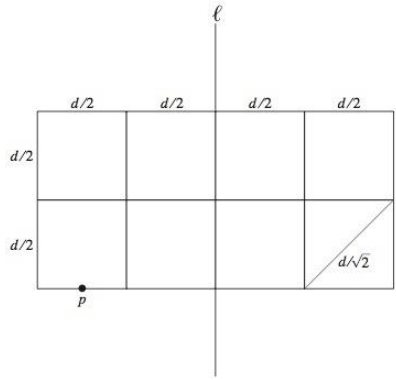
Esse problema aparece em muitas aplicações, tais como determinar o par mais próximo de aviões no espaço aéreo, em uma determinada altitude, monitorado por um controlador de tráfego aéreo. Como podemos encontrar esse par mais próximo de forma eficiente?

Solução: Para resolver esse problema, podemos primeiramente determinar a distância entre cada par de pontos e, então, encontrar a menor dessas distâncias. Entretanto, esta aproximação requer $O(n^2)$ computações de distâncias e comparações, porque há $C(n, 2) = n(n-1)/2$ pares de pontos. Surpreendentemente, há um algoritmo de divisão e resolução que pode resolver esse problema para n pontos, usando $O(n \log n)$ computações de distâncias e comparações. O algoritmo que descreveremos aqui é de autoria de Michael Samos (veja [PrSa85]).

Para simplificar, assumimos que $n = 2^k$, em que k é um número inteiro positivo. (Para evitar algumas considerações técnicas que são necessárias quando n não é uma potência de 2.) Quando $n = 2$, temos apenas um par de pontos; a distância entre esses pontos é a distância mínima. No início do algoritmo, usamos a merge sort duas vezes: uma para ordenar os pontos em ordem crescente de coordenadas x, e outra para ordenar os pontos em ordem crescente de coordenadas y. Cada uma dessas ordenações requer $O(n \log n)$ operações. Utilizaremos essa lista ordenada em cada passo recursivo.

A parte recursiva do algoritmo divide o problema em dois subproblemas, cada um envolvendo metade do número de pontos. Usando a lista ordenada de pontos a partir das coordenadas x, construímos uma linha vertical ℓ, dividindo os n pontos em duas partes, uma à esquerda e outra à direita, de tamanhos iguais, cada uma com $n/2$ pontos, como mostrado na Figura 1. (Se algum ponto estiver na linha ℓ que os divide, poderá ser usado nas duas partes, se necessário.) Na etapa subseqüente da recursão, não precisamos ordenar novamente as coordenadas x, porque podemos selecionar os subconjuntos ordenados correspondentes de todos os pontos. Esta seleção é uma tarefa que pode ser feita com $O(n)$ comparações.

Há três possibilidades que consideram as posições dos pontos mais próximos: (1) eles estão ambos na região à esquerda L, (2) eles estão ambos na região à direita R, ou (3) um ponto está à esquerda e outro à direita. Aplicar o algoritmo recursivamente para computar d_L e d_R, em que d_L é a distância mínima entre os pontos na região da esquerda e d_R é a distância mínima entre os pontos na região da direita. Considere $d = \min(d_L, d_R)$. Para dividir o problema de encontrar os dois pontos mais próximos no conjunto original em dois problemas para encontrar as menores distâncias entre dois pontos em duas regiões separadamente, temos que trabalhar com a parte da resolução do algoritmo, o que requer que consideremos o caso em que os pontos mais próximos estejam em regiões diferentes, ou seja, um ponto em L e outro em R. Como há um par de pontos com distância d, em que ambos os pontos estão em R ou em L, para os pontos mais próximos, que estão em diferentes regiões é necessário que eles estejam em uma distância menor que d.



No máximo oito pontos, incluindo p, podem estar dentro de ou sobre o retângulo $2d \times d$ centrado em ℓ, porque, pelo menos, um ponto está dentro de cada um dos oito quadrados $(d/2) \times (d/2)$.

FIGURA 2 Mostrando que Há, Pelo Menos, Outros Sete Pontos a Serem Considerados para Cada Ponto na Faixa.

Para um ponto à esquerda e um ponto à direita em uma distância menor que d, esses pontos devem estar na faixa vertical de largura $2d$ que contém a linha ℓ como seu centro. (Por outro lado, a distância entre esses pontos é maior que a distância em suas coordenadas x, a qual excede d.) Para examinar os pontos dentro da faixa, ordenamos os pontos a fim de que sejam listados em ordem crescente de coordenadas y, usando a lista ordenada dos pontos a partir de suas coordenadas y. Em cada passo recursivo, formamos um subconjunto dos pontos na região ordenada a partir de suas coordenadas y de seu conjunto já ordenado de todos os pontos, que pode ser feito com $O(n)$ comparações.

Começando com um ponto na faixa com a menor coordenada y, examinamos sucessivamente cada ponto na faixa, computando a distância entre este ponto e todos os outros nessa faixa que têm coordenadas y maiores que pudessem estar a uma distância menor que d a partir deste ponto. Note que para examinar um ponto p, precisamos considerar apenas as distâncias entre p e os pontos no conjunto que estão dentro do retângulo de altura d e largura $2d$, com p em sua base e com lados verticais com distância de d a ℓ.

Podemos mostrar que há, no máximo, oito pontos do conjunto, incluindo p, dentro do retângulo $2d \times d$ ou sobre ele. Para visualizar isso, note que pode haver, no máximo, um ponto em cada um dos oito quadrados $d/2 \times d/2$ mostrados na Figura 2. Isto segue do fato de que cada par de pontos dentro de ou sobre cada quadrado pode estar, no máximo, a uma distância $d/\sqrt{2}$, que é a medida da diagonal desse quadrado (a qual pode ser verificada pelo teorema de Pitágoras), essa distância é menor que d, e cada um desses quadrados $d/2 \times d/2$ está completamente dentro da região à esquerda ou à direita. Isso significa que nesta etapa precisamos comparar, no máximo, sete distâncias, entre p e os sete ou menos pontos no retângulo, com d.

Como o número total de pontos na faixa de largura $2d$ não excede a n (o número total de pontos no conjunto), no máximo $7n$ distâncias precisam ser comparadas com d para encontrar a distância mínima entre os pontos, ou seja, há apenas $7n$ distâncias possíveis que poderiam ser menores que d. Conseqüentemente, uma vez usada a merge sort para ordenar os pares de acordo com suas coordenadas x e y, encontramos que a função crescente $f(n)$ satisfaz a relação de recorrência

$$f(n) = 2f(n/2) + 7n,$$

em que $f(2) = 1$, excede o número de comparações necessárias para resolver o problema do par de pontos mais próximo para n pontos. A partir do *Master Theorem* (Teorema 2), temos que $f(n)$ é $O(n \log n)$. As duas ordenações de pontos a partir de suas coordenadas x e y podem ser feitas usando $O(n \log n)$ comparações, usando a merge sort, e os subconjuntos ordenados com essas coordenadas em cada um dos $O(\log n)$ passos do algoritmo podem ser feitos usando $O(n)$ comparações. Assim, verificamos que o problema do par de pontos mais próximo pode ser resolvido usando $O(n \log n)$ comparações. ◀

Exercícios

1. Quantas comparações são necessárias para uma busca binária em um conjunto de 64 elementos?
2. Quantas comparações são necessárias para localizar os elementos máximo e mínimo em uma seqüência com 128 elementos, usando o algoritmo dado no Exemplo 2?
3. Multiplique $(1110)_2$ por $(1010)_2$ usando o algoritmo de multiplicação rápida.
4. Expresse o algoritmo de multiplicação rápida em pseudocódigo.
5. Determine um valor para a constante C dada no Exemplo 4 e utilize-a para estimar o número de operações binárias necessárias para multiplicar dois números inteiros de 64 bits usando o algoritmo de multiplicação rápida.
6. Quantas operações são necessárias para multiplicar duas matrizes 32×32 usando o algoritmo dado no Exemplo 4?
7. Suponha que $f(n) = f(n/3) + 1$ quando n for divisível por 3, e $f(1) = 1$. Encontre
   a) $f(3)$.
   b) $f(27)$.
   c) $f(729)$.
8. Suponha que $f(n) = 2f(n/2) + 3$ quando n for par e $f(1) = 5$. Encontre
   a) $f(2)$.
   b) $f(8)$.
   c) $f(64)$.
   d) $f(1024)$.
9. Suponha que $f(n) = f(n/5) + 3n^2$ quando n for divisível por 5 e $f(1) = 4$. Encontre
   a) $f(5)$. **b)** $f(125)$. **c)** $f(3125)$.
10. Encontre $f(n)$ quando $n = 2^k$, em que f satisfaz a relação de recorrência $f(n) = f(n/2) + 1$, com $f(1) = 1$.
11. Faça a estimativa do tamanho de f no Exercício 10, se f for uma função crescente.
12. Encontre $f(n)$ quando $n = 3^k$, em que f satisfaz a relação de recorrência $f(n) = 2f(n/3) + 4$, com $f(1) = 1$.
13. Faça a estimativa do tamanho de f no Exercício 12, se f for uma função crescente.
14. Suponha que há $n = 2^k$ times em um torneio eliminatório, em que há $n/2$ partidas na primeira rodada, com $n/2 = 2^{k-1}$ vencedores jogando a segunda rodada, e assim por diante. Desenvolva uma relação de recorrência para o número de rodadas no campeonato.
15. Quantas rodadas ocorrerão na eliminatória descrita no Exercício 14, quando há 32 times?
16. Resolva a relação de recorrência para o número de rodadas no torneio descrito no Exercício 14.
17. Suponha que os votos de n pessoas para candidatos diferentes (em que pode haver mais que dois candidatos) para um determinado escritório sejam os elementos de uma seqüência. Uma pessoa vence a eleição se receber a maioria dos votos.
   a) Construa um algoritmo de divisão e resolução que determine se um candidato recebeu a maioria dos votos e, se isso ocorrer, qual é o candidato. [*Dica:* Suponha que n seja par e divida a seqüência de votos em duas seqüências, cada uma com $n/2$ elementos. Note que um candidato não pode receber a maioria dos votos sem ter a maioria em, pelo menos, uma das duas metades.]
   b) Use o *Master Theorem* para estimar o número de comparações necessárias pelo algoritmo que você construiu no item (a).
18. Suponha que cada pessoa em um grupo de n pessoas vote em exatamente duas pessoas de uma lista de candidatos para preencher duas vagas em um comitê. Os vencedores ganham o cargo contanto que recebam mais de $n/2$ votos.
   a) Construa um algoritmo de divisão e resolução que determine se os dois candidatos que receberam mais votos tiveram, pelo menos, $n/2$ votos e, se isso ocorrer, determine quais são esses candidatos.
   b) Use o *Master Theorem* para estimar o número de comparações necessárias pelo algoritmo que você construiu no item (a).
19. **a)** Encontre uma relação de recorrência de divisão e resolução para o número de multiplicações necessárias para computar x^n, em que x é um número real e n é um número inteiro positivo, usando o algoritmo recursivo do Exercício 26 da Seção 4.4.
   b) Use a relação de recorrência que você encontrou no item (a) para construir uma estimativa O grande para o número de multiplicações utilizadas para computar x^n, usando o algoritmo recursivo.
20. **a)** Encontre uma relação de recorrência de divisão e resolução para o número de multiplicações modulares necessárias para computar a^n **mod** m, em que a, m e n são números inteiros positivos, usando o algoritmo recursivo do Exemplo 3 da Seção 4.4.
   b) Use a relação de recorrência que você encontrou no item (a) para construir uma estimativa O grande para o número de multiplicações modulares utilizadas para computar a^n **mod** m usando o algoritmo recursivo.

em que $f(2) = 1$, excede o número de comparações necessárias para resolver o problema do par de pontos mais próximo para n pontos. A partir do *Master Theorem* (Teorema 2), temos que $f(n)$ é $O(n \log n)$. As duas ordenações de pontos a partir de suas coordenadas x e y podem ser feitas usando $O(n \log n)$ comparações, usando a merge sort, e os subconjuntos ordenados com essas coordenadas em cada um dos $O(\log n)$ passos do algoritmo podem ser feitos usando $O(n)$ comparações. Assim, verificamos que o problema do par de pontos mais próximo pode ser resolvido usando $O(n \log n)$ comparações. ◀

Exercícios

1. Quantas comparações são necessárias para uma busca binária em um conjunto de 64 elementos?
2. Quantas comparações são necessárias para localizar os elementos máximo e mínimo em uma seqüência com 128 elementos, usando o algoritmo dado no Exemplo 2?
3. Multiplique $(1110)_2$ por $(1010)_2$ usando o algoritmo de multiplicação rápida.
4. Expresse o algoritmo de multiplicação rápida em pseudocódigo.
5. Determine um valor para a constante C dada no Exemplo 4 e utilize-a para estimar o número de operações binárias necessárias para multiplicar dois números inteiros de 64 bits usando o algoritmo de multiplicação rápida.
6. Quantas operações são necessárias para multiplicar duas matrizes 32×32 usando o algoritmo dado no Exemplo 4?
7. Suponha que $f(n) = f(n/3) + 1$ quando n for divisível por 3, e $f(1) = 1$. Encontre
   a) $f(3)$.
   b) $f(27)$.
   c) $f(729)$.
8. Suponha que $f(n) = 2f(n/2) + 3$ quando n for par e $f(1) = 5$. Encontre
   a) $f(2)$.
   b) $f(8)$.
   c) $f(64)$.
   d) $f(1024)$.
9. Suponha que $f(n) = f(n/5) + 3n^2$ quando n for divisível por 5 e $f(1) = 4$. Encontre
   a) $f(5)$. **b)** $f(125)$. **c)** $f(3125)$.
10. Encontre $f(n)$ quando $n = 2^k$, em que f satisfaz a relação de recorrência $f(n) = f(n/2) + 1$, com $f(1) = 1$.
11. Faça a estimativa do tamanho de f no Exercício 10, se f for uma função crescente.
12. Encontre $f(n)$ quando $n = 3^k$, em que f satisfaz a relação de recorrência $f(n) = 2f(n/3) + 4$, com $f(1) = 1$.
13. Faça a estimativa do tamanho de f no Exercício 12, se f for uma função crescente.
14. Suponha que há $n = 2^k$ times em um torneio eliminatório, em que há $n/2$ partidas na primeira rodada, com $n/2 = 2^{k-1}$ vencedores jogando a segunda rodada, e assim por diante. Desenvolva uma relação de recorrência para o número de rodadas no campeonato.
15. Quantas rodadas ocorrerão na eliminatória descrita no Exercício 14, quando há 32 times?
16. Resolva a relação de recorrência para o número de rodadas no torneio descrito no Exercício 14.
17. Suponha que os votos de n pessoas para candidatos diferentes (em que pode haver mais que dois candidatos) para um determinado escritório sejam os elementos de uma seqüência. Uma pessoa vence a eleição se receber a maioria dos votos.
   a) Construa um algoritmo de divisão e resolução que determine se um candidato recebeu a maioria dos votos e, se isso ocorrer, qual é o candidato. [*Dica:* Suponha que n seja par e divida a seqüência de votos em duas seqüências, cada uma com $n/2$ elementos. Note que um candidato não pode receber a maioria dos votos sem ter a maioria em, pelo menos, uma das duas metades.]
   b) Use o *Master Theorem* para estimar o número de comparações necessárias pelo algoritmo que você construiu no item (a).
18. Suponha que cada pessoa em um grupo de n pessoas vote em exatamente duas pessoas de uma lista de candidatos para preencher duas vagas em um comitê. Os vencedores ganham o cargo contanto que recebam mais de $n/2$ votos.
   a) Construa um algoritmo de divisão e resolução que determine se os dois candidatos que receberam mais votos tiveram, pelo menos, $n/2$ votos e, se isso ocorrer, determine quais são esses candidatos.
   b) Use o *Master Theorem* para estimar o número de comparações necessárias pelo algoritmo que você construiu no item (a).
19. **a)** Encontre uma relação de recorrência de divisão e resolução para o número de multiplicações necessárias para computar x^n, em que x é um número real e n é um número inteiro positivo, usando o algoritmo recursivo do Exercício 26 da Seção 4.4.
   b) Use a relação de recorrência que você encontrou no item (a) para construir uma estimativa O grande para o número de multiplicações utilizadas para computar x^n, usando o algoritmo recursivo.
20. **a)** Encontre uma relação de recorrência de divisão e resolução para o número de multiplicações modulares necessárias para computar a^n **mod** m, em que a, m e n são números inteiros positivos, usando o algoritmo recursivo do Exemplo 3 da Seção 4.4.
   b) Use a relação de recorrência que você encontrou no item (a) para construir uma estimativa O grande para o número de multiplicações modulares utilizadas para computar a^n **mod** m usando o algoritmo recursivo.

34. Encontre $f(n)$ quando $n = 4^k$, em que f satisfaz a relação de recorrência $f(n) = 5f(n/4) + 6n$, com $f(1) = 1$.

35. Faça a estimativa do tamanho de f no Exercício 34 se f for uma função crescente.

36. Encontre $f(n)$ quando $n = 2^k$, em que f satisfaz a relação de recorrência $f(n) = 8f(n/2) + n^2$, com $f(1) = 1$.

37. Faça a estimativa do tamanho de f no Exercício 36 se f for uma função crescente.

7.4 Funções Generalizadas

Introdução

Links

Funções generalizadas são usadas para representar seqüências de forma eficiente, codificando os termos de uma seqüência como coeficientes de potências de uma variável x em uma série formal de potências. As funções generalizadas podem ser utilizadas para resolver muitos problemas de contagem, tais como o número de maneiras de escolher ou distribuir objetos de tipos diferentes, sujeitos a uma série de limitações, bem como o número de maneiras de organizar trocos usando moedas de real de diferentes valores. As funções generalizadas podem ser utilizadas para resolver relações de recorrência, transpondo uma relação de recorrência para os termos de uma seqüência em uma equação que envolve uma função generalizada. Essa equação pode, então, ser resolvida para encontrar a forma fechada para uma função generalizada. A partir dessa forma fechada, os coeficientes da série de potência para a função podem ser encontrados resolvendo-se a relação de recorrência original. As funções generalizadas podem ser utilizadas também para demonstrar identidades combinatórias, aproveitando-se de relações relativamente simples entre as funções que são transpostas em identidades que envolvem os termos das seqüências. As funções generalizadas são uma ferramenta bastante útil para estudar muitas propriedades das seqüências além daquelas descritas nesta seção, tais como estabelecer fórmulas assintóticas para os termos de uma seqüência.

Comecemos com a definição de função generalizada para uma seqüência.

DEFINIÇÃO 1

A *função generalizada para a seqüência* $a_0, a_1, \ldots, a_k, \ldots$ de números reais é a série infinita

$$G(x) = a_0 + a_1x + \cdots + a_kx^k + \cdots = \sum_{k=0}^{\infty} a_kx^k.$$

Lembre-se: A função generalizada para $\{a_k\}$ dada na Definição 1 é também chamada de **função generalizada ordinária** de $\{a_k\}$ para distingui-la de outros tipos de funções para esta seqüência.

EXEMPLO 1

Exemplos Extras

As funções generalizadas para as seqüências $\{a_k\}$, com $a_k = 3$, $a_k = k + 1$ e $a_k = 2^k$, são $\sum_{k=0}^{\infty} 3x^k$, $\sum_{k=0}^{\infty} (k+1)x^k$ e $\sum_{k=0}^{\infty} 2^k x^k$, respectivamente. ◀

Podemos definir as funções generalizadas para seqüências finitas de números reais estendendo uma seqüência finita $a_0, a_1, \ldots, a_n$ para uma seqüência infinita por meio do acréscimo de $a_{n+1} = 0$, $a_{n+2} = 0$, e assim por diante. A função generalizada $G(x)$ desta seqüência infinita $\{a_n\}$ é um polinômio de grau n, pois nenhum termo na forma $a_j x^j$, com $j > n$, aparece na soma final, ou seja,

$$G(x) = a_0 + a_1x + \cdots + a_nx^n.$$

EXEMPLO 2 Qual é a função generalizada para a seqüência 1, 1, 1, 1, 1, 1?

Solução: A função generalizada de 1, 1, 1, 1, 1, 1 é

$$1 + x + x^2 + x^3 + x^4 + x^5.$$

A partir do Teorema 1 da Seção 2.4, temos

$$(x^6 - 1)/(x - 1) = 1 + x + x^2 + x^3 + x^4 + x^5$$

quando $x \neq 1$. Conseqüentemente, $G(x) = (x^6 - 1)/(x - 1)$ é a função generalizada da seqüência 1, 1, 1, 1, 1, 1. [Como as potências de x são apenas detentoras de lugares para os termos da seqüência em uma função generalizada, não precisamos nos preocupar com o fato de $G(1)$ não ser definida.] ◀

EXEMPLO 3 Considere m como um número inteiro positivo. Considere $a_k = C(m, k)$, para $k = 0, 1, 2, \ldots, m$. Qual é a função generalizada da seqüência $a_0, a_1, \ldots, a_m$?

Solução: A função generalizada para essa seqüência é

$$G(x) = C(m, 0) + C(m, 1)x + C(m, 2)x^2 + \cdots + C(m,m)x^m.$$

O Teorema Binomial mostra que $G(x) = (1 + x)^m$. ◀

Fatos Úteis sobre Séries de Potências

Quando as funções generalizadas são utilizadas para resolver problemas de contagem, elas são normalmente consideradas como **séries de potência formais.** Algumas questões sobre convergência dessas séries são ainda ignoradas. Entretanto, para aplicar alguns resultados do cálculo, muitas vezes é importante considerar, para cada x, a convergência da série de potência. O fato de cada função ter uma única série de potência em torno de $x = 0$ será também importante. Em geral, entretanto, não nos preocuparemos com questões de convergência ou unicidade de séries de potência em nossas discussões. Os leitores familiarizados com cálculo podem consultar a literatura sobre esse assunto para obter detalhes sobre séries de potências, incluindo a convergência das séries que considerarmos aqui.

Agora estabeleceremos alguns fatos importantes sobre séries infinitas usadas quando trabalhamos com funções generalizadas. Uma discussão sobre resultados como estes ou a eles relacionados pode ser encontrada em livros teóricos de cálculo.

EXEMPLO 4 A função $f(x) = 1/(1 - x)$ é a função generalizada da seqüência 1, 1, 1, 1, . . ., pois

$$1/(1 - x) = 1 + x + x^2 + \cdots$$

para $|x| < 1$. ◀

EXEMPLO 5 A função $f(x) = 1/(1 - ax)$ é a função generalizada da seqüência $1, a, a^2, a^3, \ldots$, pois

$$1/(1 - ax) = 1 + ax + a^2x^2 + \cdots$$

quando $|ax| < 1$ ou, equivalentemente, para $|x| < 1/|a|$, com $a \neq 0$. ◀

Necessitamos também de alguns resultados sobre como adicionar e multiplicar duas funções generalizadas. As demonstrações desses resultados podem ser encontradas em livros teóricos de cálculo.

TEOREMA 1

Considere $f(x) = \sum_{k=0}^{\infty} a_k x^k$ e $g(x) = \sum_{k=0}^{\infty} b_k x^k$. Então

$$f(x) + g(x) = \sum_{k=0}^{\infty} (a_k + b_k) x^k \quad \text{e} \quad f(x)g(x) = \sum_{k=0}^{\infty} \left(\sum_{j=0}^{k} a_j b_{k-j} \right) x^k.$$

Lembre-se: O Teorema 1 é válido apenas para séries de potências que convergem em um intervalo, como ocorre com todas as séries apresentadas nesta seção. Entretanto, a teoria das funções generalizadas não é limitada a tais séries. Nesse caso, de séries que não convergem, as afirmações do Teorema 1 podem ser consideradas como definições de adição e multiplicação de funções generalizadas.

Ilustraremos como o Teorema 1 pode ser usado com o Exemplo 6.

EXEMPLO 6 Considere $f(x) = 1/(1-x)^2$. Use o Exemplo 4 para encontrar os coeficientes $a_0, a_1, a_2, \ldots$ na expansão $f(x) = \sum_{k=0}^{\infty} a_k x^k$.

Solução: A partir do Exemplo 4, vemos que

$$1/(1-x) = 1 + x + x^2 + x^3 + \cdots.$$

Assim, a partir do Teorema 1, temos

$$1/(1-x)^2 = \sum_{k=0}^{\infty} \left(\sum_{j=0}^{k} 1 \right) x^k = \sum_{k=0}^{\infty} (k+1) x^k.$$

◀

Lembre-se: Este resultado pode ser também derivado a partir do Exemplo 4 por diferenciação. Derivar é uma técnica útil para produzir novas identidades a partir das funções generalizadas existentes.

Para utilizar as funções generalizadas na resolução de muitos problemas importantes de contagem, precisaremos também aplicar o Teorema Binomial para expoentes que não são números inteiros positivos. Antes de apresentarmos uma versão estendida do Teorema Binomial, precisamos definir coeficientes binomiais estendidos.

DEFINIÇÃO 2

Considere u como um número real e k como um número inteiro não negativo. Então, o *coeficiente binomial estendido* $\binom{u}{k}$ é definido por

$$\binom{u}{k} = \begin{cases} u(u-1)\cdots(u-k+1)/k! & \text{se } k > 0, \\ 1 & \text{se } k = 0. \end{cases}$$

EXEMPLO 7 Encontre os valores dos coeficientes binomiais estendidos $\binom{-2}{3}$ e $\binom{1/2}{3}$.

Solução: Considerando $u = -2$ e $k = 3$ como na Definição 2, temos

$$\binom{-2}{3} = \frac{(-2)(-3)(-4)}{3!} = -4.$$

Da mesma forma, considere $u = 1/2$ e $k = 3$. Temos

$$\begin{aligned}\binom{1/2}{3} &= \frac{(1/2)(1/2-1)(1/2-2)}{3!} \\ &= (1/2)(-1/2)(-3/2)/6 \\ &= 1/16.\end{aligned}$$

◀

O Exemplo 8 fornece uma fórmula útil para coeficientes binomiais estendidos quando o parâmetro máximo é um número inteiro negativo. Ela será útil em nossas discussões subseqüentes.

EXEMPLO 8 Quando o parâmetro máximo é um número inteiro negativo, o coeficiente binomial estendido pode ser expresso em termos de um coeficiente binomial ordinário. Para ver se este é o caso, note que

$$\begin{aligned}\binom{-n}{r} &= \frac{(-n)(-n-1)\cdots(-n-r+1)}{r!} && \text{pela definição de coeficiente binomial estendido} \\ &= \frac{(-1)^r n(n+1)\cdots(n+r-1)}{r!} && \text{pondo em evidência } -1 \text{ de cada termo do numerador} \\ &= \frac{(-1)^r (n+r-1)(n+r-2)\cdots n}{r!} && \text{pela lei comutativa para multiplicação} \\ &= \frac{(-1)^r (n+r-1)!}{r!(n-1)!} && \text{multiplicando o numerador e o denominador por } (n-1)! \\ &= (-1)^r \binom{n+r-1}{r} && \text{pela definição de coeficientes binomiais} \\ &= (-1)^r C(n+r-1, r). && \text{usando a notação alternativa para coeficientes binomiais}\end{aligned}$$

◀

Estabelecemos agora o Teorema Binomial estendido.

TEOREMA 2

TEOREMA BINOMIAL ESTENDIDO Considere x como um número real com $|x| < 1$ e considere u como um número real. Então,

$$(1+x)^u = \sum_{k=0}^{\infty} \binom{u}{k} x^k.$$

O Teorema 2 pode ser obtido usando a teoria das séries de Maclaurin. Deixamos essa demonstração para o leitor familiarizado com essa parte de cálculo.

Lembre-se: Quando u for um número inteiro positivo, o Teorema Binomial estendido se reduz ao Teorema Binomial apresentado na Seção 5.4, pois, nesse caso, $\binom{u}{k} = 0$ se $k > u$.

O Exemplo 9 ilustra o uso do Teorema 2 quando o expoente é um número inteiro negativo.

EXEMPLO 9 Encontre as funções generalizadas para $(1 + x)^{-n}$ e $(1 - x)^{-n}$, em que n é um número inteiro positivo, usando o Teorema Binomial estendido.

Solução: A partir do Teorema Binomial estendido, temos que

$$(1+x)^{-n} = \sum_{k=0}^{\infty} \binom{-n}{k} x^k.$$

Usando o Exemplo 8, que fornece uma fórmula simples para $\binom{-n}{k}$, obtemos

$$(1+x)^{-n} = \sum_{k=0}^{\infty} (-1)^k C(n+k-1, k) x^k.$$

Substituindo x por $-x$, encontramos que

$$(1-x)^{-n} = \sum_{k=0}^{\infty} C(n+k-1, k) x^k.$$

◀

A Tabela 1 apresenta um sumário útil para funções generalizadas que aparecem com freqüência.

Lembre-se: Note que a segunda e a terceira fórmula dessa tabela podem ser deduzidas a partir da primeira fórmula, substituindo ax e x^r por x, respectivamente. Da mesma forma, a sexta e a sétima fórmulas podem ser deduzidas a partir da quinta, usando as mesmas substituições. A décima e a décima primeira podem ser deduzidas a partir da nona fórmula, substituindo $-x$ e ax por x, respectivamente. Algumas das fórmulas dessa tabela também podem ser derivadas a partir de outras fórmulas usando métodos de cálculo (tais como diferenciação e integração). Os estudantes devem sentir-se encorajados a saber as fórmulas principais dessa tabela (ou seja, as fórmulas a partir das quais outras podem ser derivadas, talvez a primeira, quarta, quinta, oitava, nona, décima segunda e décima terceira) e entender como derivar outras fórmulas a partir dessas fórmulas-base.

Problemas de Contagem e Funções Generalizadas

As funções generalizadas podem ser usadas para resolver uma grande variedade de problemas de contagem. Em particular, elas podem ser utilizadas para contar o número de combinações de diferentes tipos. No Capítulo 5, desenvolvemos técnicas para contar r-combinações a partir de um conjunto com n elementos com possibilidade de repetição e restrições adicionais. Tais problemas são equivalentes a contar as soluções para as equações na forma

$$e_1 + e_2 + \cdots + e_n = C,$$

em que C é uma constante e cada e_i é um número inteiro não negativo que pode ser sujeito a determinada restrição. As funções generalizadas também podem ser usadas para resolver problemas de contagem deste tipo, como mostram os exemplos 10 a 12.

EXEMPLO 10 Encontre o número de soluções de

$$e_1 + e_2 + e_3 = 17,$$

em que e_1, e_2 e e_3 são números inteiros não negativos, com $2 \leq e_1 \leq 5$, $3 \leq e_2 \leq 6$ e $4 \leq e_3 \leq 7$.

TABELA 1 Funções Generalizadas Úteis.

$G(x)$	a_k
$(1+x)^n = \sum_{k=0}^{n} C(n,k)x^k = 1 + C(n,1)x + C(n,2)x^2 + \cdots + x^n$	$C(n,k)$
$(1+ax)^n = \sum_{k=0}^{n} C(n,k)a^k x^k = 1 + C(n,1)ax + C(n,2)a^2x^2 + \cdots + a^n x^n$	$C(n,k)a^k$
$(1+x^r)^n = \sum_{k=0}^{n} C(n,k)x^{rk} = 1 + C(n,1)x^r + C(n,2)x^{2r} + \cdots + x^{rn}$	$C(n,k/r)$ se $r \mid k$; 0 caso contrário
$\frac{1-x^{n+1}}{1-x} = \sum_{k=0}^{n} x^k = 1 + x + x^2 + \cdots + x^n$	1 se $k \le n$; 0 caso contrário
$\frac{1}{1-x} = \sum_{k=0}^{\infty} x^k = 1 + x + x^2 + \cdots$	1
$\frac{1}{1-ax} = \sum_{k=0}^{\infty} a^k x^k = 1 + ax + a^2x^2 + \cdots$	a^k
$\frac{1}{1-x^r} = \sum_{k=0}^{\infty} x^{rk} = 1 + x^r + x^{2r} + \cdots$	1 se $r \mid k$; 0 caso contrário
$\frac{1}{(1-x)^2} = \sum_{k=0}^{\infty} (k+1)x^k = 1 + 2x + 3x^2 + \cdots$	$k+1$
$\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)x^k = 1 + C(n,1)x + C(n+1,2)x^2 + \cdots$	$C(n+k-1,k) = C(n+k-1,n-1)$
$\frac{1}{(1+x)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)(-1)^k x^k = 1 - C(n,1)x + C(n+1,2)x^2 - \cdots$	$(-1)^k C(n+k-1,k) = (-1)^k C(n+k-1,n-1)$
$\frac{1}{(1-ax)^n} = \sum_{k=0}^{\infty} C(n+k-1,k)a^k x^k = 1 + C(n,1)ax + C(n+1,2)a^2x^2 + \cdots$	$C(n+k-1,k)a^k = C(n+k-1,n-1)a^k$
$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$	$1/k!$
$\ln(1+x) = \sum_{k=0}^{\infty} \frac{(-1)^{k+1}}{k} x^k = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots$	$(-1)^{k+1}/k$

Nota: As séries para as duas últimas funções generalizadas podem ser encontradas na maioria dos livros teóricos de cálculo quando são discutidas séries de potência.

Solução: O número de soluções com as restrições indicadas é o coeficiente de x^{17} na expansão de

$$(x^2 + x^3 + x^4 + x^5)(x^3 + x^4 + x^5 + x^6)(x^4 + x^5 + x^6 + x^7).$$

Isto ocorre porque obtemos um termo igual a x^{17} no produto, escolhendo um termo na primeira soma x^{e_1}, um termo na segunda soma x^{e_2} e um termo na terceira soma x^{e_3}, em que os expoentes e_1, e_2 e e_3 satisfazem a equação $e_1 + e_2 + e_3 = 17$ e as restrições dadas.

Não é difícil ver que o coeficiente de x^{17} neste produto é 3. Assim, há três soluções. (Note que o cálculo desse coeficiente envolve o trabalho de enumerar todas as soluções das equações com as restrições dadas. Entretanto, o método que ilustramos pode ser utilizado como método geral para resolver uma grande classe de problemas de contagem com fórmulas especiais, como veremos. Além disso, um sistema de álgebra computacional pode ser utilizado para tais computações.) ◀

EXEMPLO 11 De quantas maneiras diferentes oito biscoitos idênticos podem ser distribuídos entre três crianças distintas, se cada criança receber, pelo menos, dois biscoitos e não mais que quatro?

Solução: Como cada criança recebe, pelo menos, dois biscoitos, mas não mais que quatro, para cada criança há um fator igual a

$$(x^2 + x^3 + x^4)$$

na função generalizada para a seqüência $\{c_n\}$, em que c_n é o número de maneiras de distribuir n biscoitos. Como há três crianças, esta função generalizada é

$$(x^2 + x^3 + x^4)^3.$$

Precisamos do coeficiente de x^8 neste produto. A razão é que os termos x^8 na expansão correspondem às maneiras em que os três termos podem ser escolhidos, com um de cada fator, que tem expoentes que somam 8. Além disso, os expoentes do termo do primeiro, do segundo e do terceiro fator são os números de biscoitos que a primeira, a segunda e a terceira criança recebem, respectivamente. A computação dessa potência mostra que este coeficiente é igual a 6. Assim, há seis maneiras de distribuir os biscoitos a fim de que cada criança receba, pelo menos, dois, mas não mais que quatro biscoitos. ◀

EXEMPLO 12 Use as funções generalizadas para determinar o número de maneiras de inserir fichas que valem R$ 1,00, R$ 2,00 e R$ 5,00 em uma máquina eletrônica para pagar por um produto que custa r reais, quando a ordem na qual as fichas são inseridas é relevante e quando a ordem não é relevante. (Por exemplo, há duas maneiras de pagar por um produto que custa R$ 3,00, quando a ordem na qual as fichas são inseridas não é relevante: inserindo três fichas de R$ 1,00 ou uma de R$ 1,00 e uma de R$ 2,00. Quando a ordem é relevante, há três maneiras: inserir três fichas de R$ 1,00, inserir uma de R$ 1,00 e então uma de R$ 2,00 ou inserir uma de R$ 2,00 e então uma de R$ 1,00.)

Solução: Considere o caso no qual a ordem em que as fichas são inseridas não é relevante. Aqui, o que nos preocupa é o número de fichas de cada tipo que são usadas para produzir um total de r reais. Como podemos usar qualquer número de fichas de R$ 1,00, de R$ 2,00 e de R$ 5,00, a resposta é o coeficiente de x^r na função generalizada

$$(1 + x + x^2 + x^3 + \cdots)(1 + x^2 + x^4 + x^6 + \cdots)(1 + x^5 + x^{10} + x^{15} + \cdots).$$

(O primeiro fator neste produto representa as fichas de R$ 1,00 usadas; o segundo, as de R$ 2,00; e o terceiro, as de R$ 5,00.) Por exemplo, o número de maneiras de pagar por um produto que custa R$ 7,00, usando fichas de R$ 1,00, R$ 2,00 e R$ 5,00, é dado pelo coeficiente de x^7 na expansão, que é igual a 6.

Quando a ordem na qual as fichas são inseridas é relevante, o número de maneiras de inserir n fichas para produzir um total de r reais é o coeficiente de x^r em

$$(x + x^2 + x^5)^n,$$

porque cada uma das r fichas podem ser de R$ 1,00, de R$ 2,00 ou de R$ 5,00. Como qualquer número de fichas pode ser inserido, o número de maneiras de produzir r reais usando fichas de R$ 1,00, de R$ 2,00 ou de R$ 5,00, quando a ordem de inserção é relevante, é o coeficiente de x^r em

$$1 + (x + x^2 + x^5) + (x + x^2 + x^5)^2 + \cdots = \frac{1}{1 - (x + x^2 + x^5)} = \frac{1}{1 - x - x^2 - x^5},$$

em que adicionamos o número de maneiras de inserir 0, 1, 2, 3 fichas, e assim por diante, e em que usamos a identidade $1/(1 - x) = 1 + x + x^2 + \cdots$, com x substituído por $x + x^2 + x^5$. Por exemplo, o número de maneiras de pagar por um produto que custa R$ 7,00 usando fichas de R$ 1,00, R$ 2,00 e R$ 5,00, quando a ordem é relevante, é o coeficiente de x^7 nesta expansão, que é igual a 26. [*Dica:* Para ver que este coeficiente é igual a 26, requer-se a adição dos coeficientes de x^7 na expansão $(x + x^2 + x^5)^k$ para $2 \leq k \leq 7$. Isso pode ser feito com um pouco de computação, ou com um sistema de álgebra computacional.] ◀

O Exemplo 13 mostra a versatilidade das funções generalizadas quando usadas para resolver problemas com diferentes hipóteses.

EXEMPLO 13 Use as funções generalizadas para encontrar o número de k-combinações de um conjunto com n elementos. Assuma que o Teorema Binomial já tenha sido demonstrado.

Solução: Cada um dos n elementos no conjunto contém o termo $(1 + x)$ para a função generalizada $f(x) = \sum_{k=0}^{n} a_k x^k$. Aqui, $f(x)$ é a função generalizada para $\{a_k\}$, em que a_k representa o número de k-combinações de um conjunto com n elementos. Assim,

$$f(x) = (1 + x)^n.$$

Mas, pelo Teorema Binomial, temos

$$f(x) = \sum_{k=0}^{n} \binom{n}{k} x^k,$$

em que

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Assim, $C(n, k)$, o número de k-combinações de um conjunto com n elementos, é

$$\frac{n!}{k!(n-k)!}.$$

◀

Lembre-se: Demonstramos o Teorema Binomial na Seção 5.4 usando a fórmula para o número de r-combinações de um conjunto com n elementos. Este exemplo mostra que o Teorema Binomial, que pode ser demonstrado por indução matemática, pode ser usado para derivar a fórmula para o número de r-combinações de um conjunto com n elementos.

EXEMPLO 14 Use as funções generalizadas para encontrar o número de r-combinações com repetição para um conjunto com n elementos.

Solução: Considere $G(x)$ como a função generalizada para a seqüência $\{a_r\}$, em que a_r é igual ao número de r-combinações com repetição de um conjunto com n elementos, ou seja, $G(x) = \sum_{r=0}^{\infty} a_r x^r$. Como podemos selecionar qualquer número de determinado membro do conjunto com n elementos quando formamos uma r-combinação com repetição, cada um dos n elementos contribui com $(1 + x + x^2 + x^3 + \cdots)$ para uma expansão do produto de $G(x)$. Cada elemento contribui com este fator porque pode ser selecionado zero vezes, uma, duas, três vezes, e assim por diante, quando uma r-combinação é formada (com um total de r elementos selecionados). Como há n elementos no conjunto e cada um contribui com o mesmo fator para $G(x)$, temos

$$G(x) = (1 + x + x^2 + \cdots)^n.$$

Enquanto $|x| < 1$, temos $1 + x + x^2 + \cdots = 1/(1 - x)$, assim

$$G(x) = 1/(1 - x)^n = (1 - x)^{-n}.$$

Aplicando o Teorema Binomial estendido (Teorema 2), temos que

$$(1 - x)^{-n} = (1 + (-x))^{-n} = \sum_{r=0}^{\infty} \binom{-n}{r} (-x)^r.$$

O número de r-combinações com repetições de um conjunto com n elementos, quando r for um número inteiro positivo, é o coeficiente a_r de x^r nesta soma. Conseqüentemente, usando o Exemplo 8, encontramos que a_r é igual a

$$\begin{aligned} \binom{-n}{r}(-1)^r &= (-1)^r C(n + r - 1, r) \cdot (-1)^r \\ &= C(n + r - 1, r). \end{aligned}$$

◀

Note que o resultado do Exemplo 14 é o mesmo que estabelecemos como Teorema 2 na Seção 5.5.

EXEMPLO 15 Use as funções generalizadas para encontrar o número de maneiras de selecionar r objetos de n tipos diferentes, se devemos selecionar, pelo menos, um objeto de cada tipo.

Solução: Como precisamos escolher, pelo menos, um objeto de cada tipo, cada um dos n tipos de objetos contribui com o fator $(x + x^2 + x^3 + \cdots)$ na função generalizada $G(x)$ para a seqüência $\{a_r\}$, em que a_r é o número de maneiras de selecionar r objetos de n tipos diferentes, se precisamos de, pelo menos, um objeto de cada tipo. Assim,

$$G(x) = (x + x^2 + x^3 + \cdots)^n = x^n(1 + x + x^2 + \cdots)^n = x^n/(1 - x)^n.$$

Usando o Teorema Binomial estendido e o Exemplo 8, temos

$$\begin{aligned} G(x) &= x^n/(1-x)^n \\ &= x^n \cdot (1-x)^{-n} \\ &= x^n \sum_{r=0}^{\infty} \binom{-n}{r} (-x)^r \\ &= x^n \sum_{r=0}^{\infty} (-1)^r C(n+r-1, r)(-1)^r x^r \\ &= \sum_{r=0}^{\infty} C(n+r-1, r) x^{n+r} \\ &= \sum_{t=n}^{\infty} C(t-1, t-n) x^t \\ &= \sum_{r=n}^{\infty} C(r-1, r-n) x^r. \end{aligned}$$

Dividimos a somatória da penúltima equação substituindo $t = n + r$ para que $t = n$ quando $r = 0$ e $n + r - 1 = t - 1$, e então substituímos t por r como o índice da somatória na última equação para retornar à nossa notação original. Assim, há $C(r-1, r-n)$ maneiras de selecionar r objetos de n tipos diferentes, se devemos selecionar, pelo menos, um objeto de cada tipo. ◀

Usando Funções Generalizadas para Resolver Relações de Recorrência

Podemos encontrar a solução para uma relação de recorrência e suas condições iniciais com uma fórmula explícita para as funções generalizadas associadas. Isso será ilustrado nos exemplos 16 e 17.

EXEMPLO 16 Resolva a relação de recorrência $a_k = 3a_{k-1}$ para $k = 1, 2, 3, \ldots$ com condição inicial $a_0 = 2$.

Exemplos Extras

Solução: Considere $G(x)$ como a função generalizada para a seqüência $\{a_k\}$, ou seja, $G(x) = \sum_{k=0}^{\infty} a_k x^k$. Primeiro note que

$$xG(x) = \sum_{k=0}^{\infty} a_k x^{k+1} = \sum_{k=1}^{\infty} a_{k-1} x^k.$$

Usando a relação de recorrência, vemos que

$$\begin{aligned} G(x) - 3xG(x) &= \sum_{k=0}^{\infty} a_k x^k - 3\sum_{k=1}^{\infty} a_{k-1} x^k \\ &= a_0 + \sum_{k=1}^{\infty} (a_k - 3a_{k-1}) x^k \\ &= 2, \end{aligned}$$

com $a_0 = 2$ e $a_k = 3a_{k-1}$. Então,

$$G(x) - 3xG(x) = (1 - 3x)G(x) = 2.$$

Resolvendo $G(x)$ vemos que $G(x) = 2/(1 - 3x)$. Usando a identidade $1/(1 - ax) = \sum_{k=0}^{\infty} a^k x^k$, da Tabela 1, temos

$$G(x) = 2\sum_{k=0}^{\infty} 3^k x^k = \sum_{k=0}^{\infty} 2 \cdot 3^k x^k.$$

Conseqüentemente, $a_k = 2 \cdot 3^k$. ◀

EXEMPLO 17 Suponha que uma senha válida é um número de n dígitos em notação decimal que contém um número par de 0s. Considere a_n como o número de senhas válidas de extensão n. No Exemplo 7 da Seção 7.1, mostramos que a seqüência $\{a_n\}$ satisfaz a relação de recorrência

$$a_n = 8a_{n-1} + 10^{n-1}$$

e a condição inicial $a_1 = 9$. Utilize as funções generalizadas para encontrar uma fórmula explícita para a_n.

Solução: Para tornar nosso trabalho com funções generalizadas mais simples, estendemos esta seqüência acrescentando $a_0 = 1$; quando determinamos esse valor para a_0 e utilizamos a relação de recorrência, temos $a_1 = 8a_0 + 10^0 = 8 + 1 = 9$, que é consistente com nossa condição inicial original. (Também faz sentido porque há um código de extensão 0 – a seqüência vazia.)

Multiplicamos os dois lados da relação de recorrência por x^n para obter

$$a_n x^n = 8a_{n-1}x^n + 10^{n-1}x^n.$$

Considere $G(x) = \sum_{n=0}^{\infty} a_n x^n$ como a função generalizada da seqüência $a_0, a_1, a_2, \ldots$. Somamos os dois lados da última equação, começando com $n = 1$, para encontrar que

$$\begin{aligned} G(x) - 1 &= \sum_{n=1}^{\infty} a_n x^n = \sum_{n=1}^{\infty} (8a_{n-1}x^n + 10^{n-1}x^n) \\ &= 8\sum_{n=1}^{\infty} a_{n-1}x^n + \sum_{n=1}^{\infty} 10^{n-1}x^n \\ &= 8x\sum_{n=1}^{\infty} a_{n-1}x^{n-1} + x\sum_{n=1}^{\infty} 10^{n-1}x^{n-1} \\ &= 8x\sum_{n=0}^{\infty} a_n x^n + x\sum_{n=0}^{\infty} 10^n x^n \\ &= 8xG(x) + x/(1-10x), \end{aligned}$$

em que usamos o Exemplo 5 para avaliar a segunda somatória. Assim, temos

$$G(x) - 1 = 8x\,G(x) + x/(1 - 10x).$$

A resolução para $G(x)$ mostra que

$$G(x) = \frac{1 - 9x}{(1 - 8x)(1 - 10x)}.$$

Expandindo o lado direito dessa equação em frações parciais (como é feito na integração de funções racionais estudadas em cálculo), temos

$$G(x) = \frac{1}{2}\left(\frac{1}{1-8x} + \frac{1}{1-10x}\right).$$

Usando o Exemplo 5 duas vezes (uma vez com $a = 8$ e outra com $a = 10$), vemos que

$$\begin{aligned} G(x) &= \frac{1}{2}\left(\sum_{n=0}^{\infty} 8^n x^n + \sum_{n=0}^{\infty} 10^n x^n\right) \\ &= \sum_{n=0}^{\infty} \frac{1}{2}(8^n + 10^n)\, x^n. \end{aligned}$$

Conseqüentemente, fica mostrado que

$$a_n = \frac{1}{2}(8^n + 10^n).$$

◀

Demonstrando Identidades via Funções Generalizadas

No Capítulo 5, vimos como identidades combinatórias podem ser estabelecidas usando demonstrações combinatórias. Aqui, mostraremos que tais identidades, assim como identidades para coeficientes binomiais estendidos, podem ser demonstradas usando as funções generalizadas. Algumas vezes, o método da função generalizada é mais simples que outros métodos, especialmente quando é mais simples trabalhar com formas fechadas de uma função generalizada do que com os termos das seqüências. Ilustraremos, com o Exemplo 18, como as funções generalizadas podem ser usadas para demonstrar identidades.

EXEMPLO 18 Utilize as funções generalizadas para mostrar que

$$\sum_{k=0}^{n} C(n, k)^2 = C(2n, n)$$

sempre que n for um número inteiro positivo.

Solução: Primeiro note que, pelo Teorema Binomial, $C(2n, n)$ é o coeficiente de x^n em $(1 + x)^{2n}$. Entretanto, temos também que

$$\begin{aligned} (1 + x)^{2n} &= [(1 + x)^n]^2 \\ &= [C(n, 0) + C(n, 1)x + C(n, 2)x^2 + \cdots + C(n, n)x^n]^2. \end{aligned}$$

O coeficiente de x^n nessa expressão é

$$C(n, 0)C(n, n) + C(n, 1)C(n, n-1) + C(n, 2)C(n, n-2) + \cdots + C(n, n)C(n, 0).$$

Isso é igual a $\sum_{k=0}^{n} C(n, k)$, pois $C(n, n-k) = C(n, k)$. Como ambos, $C(2n, n)$ e $\sum_{k=0}^{n} C(n, k)^2$ representam o coeficiente de x^n em $(1 + x)^{2n}$, eles são iguais. ◀

Os exercícios 42 e 43, no final desta seção, mostram que as identidades de Pascal e de Vandermonde podem ser demonstradas usando as funções generalizadas.

Exercícios

1. Encontre a função generalizada para a seqüência finita 2, 2, 2, 2, 2, 2.
2. Encontre a função generalizada para a seqüência finita 1, 4, 16, 64, 256.

Nos exercícios 3 a 8, entendemos como uma **forma fechada** uma expressão algébrica que não envolva uma somatória sobre um domínio de valores nem o uso de reticências.

3. Encontre uma forma fechada para a função generalizada de cada uma das seqüências abaixo. (Para cada seqüência, utilize a escolha mais óbvia de uma seqüência que contenha os termos iniciais listados.)
   a) 0, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, ...
   b) 0, 0, 0, 1, 1, 1, 1, 1, 1, ...
   c) 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, ...
   d) 2, 4, 8, 16, 32, 64, 128, 256, ...
   e) $\binom{7}{0}, \binom{7}{1}, \binom{7}{2}, \ldots, \binom{7}{7}, 0, 0, 0, 0, 0, \ldots$
   f) 2, −2, 2, −2, 2, −2, 2, −2, ...
   g) 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, ...
   h) 0, 0, 0, 1, 2, 3, 4, ...
4. Encontre uma forma fechada para a função generalizada de cada uma das seqüências abaixo. (Para cada seqüência, utilize a escolha mais óbvia de uma seqüência que contenha os termos iniciais listados.)
   a) −1, −1, −1, −1, −1, −1, −1, 0, 0, 0, 0, 0, 0, ...
   b) 1, 3, 9, 27, 81, 243, 729, ...
   c) 0, 0, 3, −3, 3, −3, 3, −3, ...
   d) 1, 2, 1, 1, 1, 1, 1, 1, 1, ...
   e) $\binom{7}{0}, 2\binom{7}{1}, 2^2\binom{7}{2}, \ldots, 2^7\binom{7}{7}, 0, 0, 0, 0, \ldots$
   f) −3, 3, −3, 3, −3, 3, ...
   g) 0, 1, −2, 4, −8, 16, −32, 64, ...
   h) 1, 0, 1, 0, 1, 0, 1, 0, ...
5. Encontre uma forma fechada para a função generalizada para a seqüência $\{a_n\}$, em que
   a) $a_n = 5$ para todo $n = 0, 1, 2, \ldots$.
   b) $a_n = 3^n$ para todo $n = 0, 1, 2, \ldots$.
   c) $a_n = 2$ para $n = 3, 4, 5, \ldots$ e $a_0 = a_1 = a_2 = 0$.
   d) $a_n = 2n + 3$ para todo $n = 0, 1, 2, \ldots$.
   e) $a_n = \binom{8}{n}$ para todo $n = 0, 1, 2, \ldots$.
   f) $a_n = \binom{n+4}{n}$ para todo $n = 0, 1, 2, \ldots$.
6. Encontre uma forma fechada para a função generalizada para a seqüência $\{a_n\}$, em que
   a) $a_n = -1$ para todo $n = 0, 1, 2, \ldots$.
   b) $a_n = 2^n$ para $n = 1, 2, 3, 4, \ldots$ e $a_0 = 0$.
   c) $a_n = n - 1$ para $n = 0, 1, 2, \ldots$.
   d) $a_n = 1/(n + 1)!$ para $n = 0, 1, 2, \ldots$.
   e) $a_n = \binom{n}{2}$ para $n = 0, 1, 2, \ldots$.
   f) $a_n = \binom{10}{n+1}$ para $n = 0, 1, 2, \ldots$.
7. Para cada uma das funções generalizadas abaixo, forneça uma fórmula fechada para a seqüência que ela determina.
   a) $(3x - 4)^3$ **b)** $(x^3 + 1)^3$
   c) $1/(1 - 5x)$ **d)** $x^3/(1 + 3x)$
   e) $x^2 + 3x + 7 + (1/(1 - x^2))$
   f) $(x^4/(1 - x^4)) - x^3 - x^2 - x - 1$
   g) $x^2/(1 - x)^2$ **h)** $2e^{2x}$
8. Para cada uma das funções generalizadas abaixo, forneça uma fórmula fechada para a seqüência que ela determina.
   a) $(x^2 + 1)^3$ **b)** $(3x - 1)^3$
   c) $1/(1 - 2x^2)$ **d)** $x^2/(1 - x)^3$
   e) $x - 1 + (1/(1 - 3x))$ **f)** $(1 + x^3)/(1 + x)^3$
   ***g)** $x/(1 + x + x^2)$ **h)** $e^{3x^2} - 1$
9. Encontre o coeficiente de x^{10} nas séries de potência para cada uma das funções abaixo.
   a) $(1 + x^5 + x^{10} + x^{15} + \cdots)^3$
   b) $(x^3 + x^4 + x^5 + x^6 + x^7 + \cdots)^3$
   c) $(x^4 + x^5 + x^6)(x^3 + x^4 + x^5 + x^6 + x^7)(1 + x + x^2 + x^3 + x^4 + \cdots)$
   d) $(x^2 + x^4 + x^6 + x^8 + \cdots)(x^3 + x^6 + x^9 + \cdots)(x^4 + x^8 + x^{12} + \cdots)$
   e) $(1 + x^2 + x^4 + x^6 + x^8 + \cdots)(1 + x^4 + x^8 + x^{12} + \cdots)(1 + x^6 + x^{12} + x^{18} + \cdots)$
10. Encontre o coeficiente de x^9 nas séries de potência para cada uma das funções abaixo.
    a) $(1 + x^3 + x^6 + x^9 + \cdots)^3$
    b) $(x^2 + x^3 + x^4 + x^5 + x^6 + \cdots)^3$
    c) $(x^3 + x^5 + x^6)(x^3 + x^4)(x + x^2 + x^3 + x^4 + \cdots)$
    d) $(x + x^4 + x^7 + x^{10} + \cdots)(x^2 + x^4 + x^6 + x^8 + \cdots)$
    e) $(1 + x + x^2)^3$
11. Encontre o coeficiente de x^{10} nas séries de potência para cada uma das funções abaixo.
    a) $1/(1 - 2x)$ **b)** $1/(1 + x)^2$
    c) $1/(1 - x)^3$ **d)** $1/(1 + 2x)^4$
    e) $x^4/(1 - 3x)^3$
12. Encontre o coeficiente de x^{12} nas séries de potência para cada uma das funções abaixo.
    a) $1/(1 + 3x)$ **b)** $1/(1 - 2x)^2$
    c) $1/(1 + x)^8$ **d)** $1/(1 - 4x)^3$
    e) $x^3/(1 + 4x)^2$
13. Utilize as funções generalizadas para determinar o número de maneiras em que 10 balões idênticos podem ser distribuídos entre quatro crianças, se cada criança receber, pelo menos, dois balões.
14. Utilize as funções generalizadas para determinar o número de maneiras em que 12 figurinhas idênticas podem ser distribuídas entre cinco crianças, para que cada criança receba, pelo menos, três figurinhas.
15. Utilize as funções generalizadas para determinar o número de maneiras em que 15 animais empalhados idênticos podem ser distribuídos entre seis crianças, para que cada uma

receba, pelo menos, um, mas não mais que três animais empalhados.

16. Utilize as funções generalizadas para determinar o número de maneiras de escolher uma dúzia de biscoitos a partir de três variedades — ovos, salgado e tradicional —, se, pelo menos, dois biscoitos de cada tipo, mas não mais que três salgados, são escolhidos.

17. De quantas maneiras 25 rosquinhas idênticas podem ser distribuídas entre quatro policiais para que cada um receba, pelo menos, três, mas não mais que sete rosquinhas?

18. Utilize as funções generalizadas para encontrar o número de maneiras de selecionar 14 bolas de um recipiente com 100 bolas vermelhas, 100 azuis e 100 verdes, para que sejam selecionadas mais que 3 e menos que 10 bolas azuis. Assuma que a ordem em que as bolas são escolhidas não é relevante.

19. Qual é a função generalizada para a seqüência $\{c_k\}$, em que c_k é o número de maneiras de organizar trocos de k reais, usando notas de R$ 1,00, R$ 2,00, R$ 5,00 e R$ 10,00?

20. Qual é a função generalizada para a seqüência $\{c_k\}$, em que c_k representa o número de maneiras de organizar trocos para k pesos usando notas de 10, 20, 50 e 100 pesos?

21. Dê uma interpretação combinatória do coeficiente de x^4 na expansão $(1 + x + x^2 + x^3 + \cdots)^3$. Utilize essa interpretação para encontrar esse número.

22. Dê uma interpretação combinatória do coeficiente de x^6 na expansão $(1 + x + x^2 + x^3 + \cdots)^n$. Utilize essa interpretação para encontrar esse número.

23. **a)** Qual é a função generalizada para $\{a_k\}$, em que a_k é o número de soluções de $x_1 + x_2 + x_3 = k$ quando x_1, x_2 e x_3 forem números inteiros com $x_1 \geq 2$, $0 \leq x_2 \leq 3$ e $2 \leq x_3 \leq 5$?

b) Utilize sua resposta do item (a) para encontrar a_6.

24. **a)** Qual é a função generalizada para $\{a_k\}$, em que a_k é o número de soluções de $x_1 + x_2 + x_3 + x_4 = k$ quando x_1, x_2, x_3 e x_4 forem números inteiros com $x_1 \geq 3$, $1 \leq x_2 \leq 5$, $0 \leq x_3 \leq 4$ e $x_4 \geq 1$?

b) Utilize sua resposta do item (a) para encontrar a_7.

25. Explique como as funções generalizadas podem ser utilizadas para encontrar o número de maneiras de uma postagem de r centavos ser feita em um envelope, usando selos de 3, 4 e 20 centavos.

a) Suponha que a ordem em que os selos são utilizados não é relevante.

b) Suponha que os selos são postos em uma fila e que a ordem em que aparecem é relevante.

c) Utilize sua resposta do item (a) para determinar o número de maneiras de uma postagem de 46 centavos ser utilizada em um envelope, usando selos de 3, 4 e 20 centavos, quando a ordem dos selos não for relevante. (É recomendada a utilização de um programa de álgebra computacional.)

d) Utilize sua resposta do item (b) para determinar o número de maneiras de uma postagem de 46 centavos ser posta em fila em um envelope, utilizando selos de 3, 4 e 20 centavos, quando a ordem for relevante. (É recomendada a utilização de um programa de álgebra computacional.)

26. **a)** Mostre que $1/(1 - x - x^2 - x^3 - x^4 - x^5 - x^6)$ é a função generalizada para o número de maneiras de obter a soma de n quando um dado é jogado repetidamente e a ordem dos lançamentos é relevante.

b) Utilize o item (a) para encontrar o número de maneiras de obter 8 quando um dado é lançado repetidamente e a ordem dos lançamentos é relevante. (É recomendada a utilização de um pacote de álgebra computacional.)

27. Utilize funções generalizadas (e um pacote de álgebra computacional, se estiver disponível) para encontrar o número de maneiras de organizar trocos para R$ 1,00 usando

a) moedas de 10 e 25 centavos.

b) moedas de 5, 10 e 25 centavos.

c) moedas de 1, 10 e 25 centavos.

d) moedas de 1, 5, 10 e 25 centavos.

28. Utilize funções generalizadas (e um pacote de álgebra computacional, se estiver disponível) para encontrar o número de maneiras de organizar o troco para R$ 1,00 usando moedas de 1, 5, 10 e 25 centavos com

a) não mais que 10 moedas de 1 centavo.

b) não mais que 10 moedas de 1 centavo e 10 moedas de 5 centavos.

***c)** não mais que 10 moedas.

29. Utilize funções generalizadas para encontrar o número de maneiras de organizar o troco para R$ 100,00 usando

a) notas de R$ 10,00, R$ 20,00 e R$ 50,00.

b) notas de R$ 5,00, R$ 10,00, R$ 20,00 e R$ 50,00.

c) notas de R$ 5,00, R$ 10,00, R$ 20,00 e R$ 50,00, se, pelo menos, uma nota de cada valor for utilizada.

d) notas de R$ 5,00, R$ 10,00 e R$ 20,00, se, pelo menos, uma e não mais que quatro notas de cada valor forem utilizadas.

30. Se $G(x)$ é a função generalizada para a seqüência $\{a_k\}$, qual é a função generalizada para cada uma das seqüências abaixo?

a) $2a_0, 2a_1, 2a_2, 2a_3, \ldots$

b) $0, a_0, a_1, a_2, a_3, \ldots$ (supondo que os termos seguem o primeiro termo)

c) $0, 0, 0, 0, a_2, a_3, \ldots$ (supondo que os termos seguem os quatro primeiros termos)

d) $a_2, a_3, a_4, \ldots$

e) $a_1, 2a_2, 3a_3, 4a_4, \ldots$ [*Dica: Necessário conhecimento de cálculo.*]

f) $a_0^2, 2a_0a_1, a_1^2 + 2a_0a_2, 2a_0a_3 + 2a_1a_2, 2a_0a_4 + 2a_1a_3 + a_2^2, \ldots$

31. Se $G(x)$ é a função generalizada para a seqüência $\{a_k\}$, qual é a função generalizada para cada uma das seqüências abaixo?

a) $0, 0, 0, a_3, a_4, a_5, \ldots$ (supondo que os termos seguem os três primeiros termos)

b) $a_0, 0, a_1, 0, a_2, 0, \ldots$

c) $0, 0, 0, 0, a_0, a_1, a_2, \ldots$ (supondo que os termos seguem os quatro primeiros termos)

d) $a_0, 2a_1, 4a_2, 8a_3, 16a_4, \ldots$

e) $0, a_0, a_1/2, a_2/3, a_3/4, \ldots$ [*Dica: Necessário conhecimento de cálculo.*]

f) $a_0, a_0 + a_1, a_0 + a_1 + a_2, a_0 + a_1 + a_2 + a_3, \ldots$

32. Utilize funções generalizadas para resolver a relação de recorrência $a_k = 7a_{k-1}$ com a condição inicial $a_0 = 5$.

33. Utilize funções generalizadas para resolver a relação de recorrência $a_k = 3a_{k-1} + 2$ com a condição inicial $a_0 = 1$.

34. Utilize funções generalizadas para resolver a relação de recorrência $a_k = 3a_{k-1} + 4^{k-1}$ com a condição inicial $a_0 = 1$.

35. Utilize funções generalizadas para resolver a relação de recorrência $a_k = 5a_{k-1} - 6a_{k-2}$ com condições iniciais $a_0 = 6$ e $a_1 = 30$.

36. Utilize funções generalizadas para resolver a relação de recorrência $a_k = a_{k-1} + 2a_{k-2} + 2^k$ com condições iniciais $a_0 = 4$ e $a_1 = 12$.

37. Utilize funções generalizadas para resolver a relação de recorrência $a_k = 4a_{k-1} - 4a_{k-2} + k^2$ com condições iniciais $a_0 = 2$ e $a_1 = 5$.

38. Utilize funções generalizadas para resolver a relação de recorrência $a_k = 2a_{k-1} + 3a_{k-2} + 4^k + 6$ com condições iniciais $a_0 = 20$, $a_1 = 60$.

39. Utilize funções generalizadas para encontrar uma fórmula explícita para os números de Fibonacci.

***40.** **a)** Mostre que, se n for um número inteiro positivo, então

$$\binom{-1/2}{n} = \frac{\binom{2n}{n}}{(-4)^n}.$$

b) Utilize a extensão do Teorema Binomial e o item (a) para mostrar que o coeficiente de x^n na expansão de $(1-4x)^{-1/2}$ é $\binom{2n}{n}$ para todos os números inteiros não negativos n.

***41.** (*Requer cálculo*) Considere $\{C_n\}$ como a seqüência de números de Catalan, ou seja, a solução da relação de recorrência $C_n = \sum_{k=0}^{n-1} C_k C_{n-k-1}$, com $C_0 = C_1 = 1$ (veja o Exemplo 8 na Seção 7.1).

a) Mostre que, se $G(x)$ for a função generalizada para a seqüência de números de Catalan, então $xG(x)^2 - G(x) + 1 = 0$. Conclua (usando as condições iniciais) que $G(x) = (1 - \sqrt{1-4x})/(2x)$.

b) Utilize o Exercício 40 para concluir que

$$G(x) = \sum_{n=0}^{\infty} \frac{1}{n+1}\binom{2n}{n} x^n,$$

em que

$$C_n = \frac{1}{n+1}\binom{2n}{n}.$$

42. Utilize funções generalizadas para demonstrar a identidade de Pascal: $C(n, r) = C(n-1, r) + C(n-1, r-1)$ quando n e r forem números inteiros positivos com $r < n$. [*Dica:* Use a identidade $(1+x)^n = (1+x)^{n-1} + x(1+x)^{n-1}$.]

43. Utilize funções generalizadas para demonstrar a identidade de Vandermonde: $C(m+n, r) = \sum_{k=0}^{r} C(m, r-k)C(n, k)$, sempre que m, n e r forem números inteiros não negativos com r não excedente a m ou n. [*Dica:* Olhe para o coeficiente de x^r nos dois lados de $(1+x)^{m+n} = (1+x)^m(1+x)^n$.]

44. Este exercício mostra como utilizar as funções generalizadas para derivar uma fórmula para a soma dos primeiros n quadrados.

a) Mostre que $(x^2 + x)/(1-x)^4$ é a função generalizada para a seqüência $\{a_n\}$, em que $a_n = 1^2 + 2^2 + \cdots + n^2$.

b) Use o item (a) para encontrar uma fórmula explícita para a soma $1^2 + 2^2 + \cdots + n^2$.

A **função exponencial generalizada** para a seqüência $\{a_n\}$ é a série

$$\sum_{n=0}^{\infty} \frac{a_n}{n!} x^n.$$

Por exemplo, a função exponencial generalizada para a seqüência $1, 1, 1, \ldots$ é a função $\sum_{n=0}^{\infty} x^n/n! = e^x$. (Você verá que esta série será útil nos exercícios abaixo.) Note que e^x é a função generalizada (ordinária) para a seqüência $1, 1, 1/2!, 1/3!, 1/4!, \ldots$.

45. Encontre uma forma fechada para a função exponencial generalizada para a seqüência $\{a_n\}$, em que

a) $a_n = 2$. **b)** $a_n = (-1)^n$.
c) $a_n = 3^n$. **d)** $a_n = n + 1$.
e) $a_n = 1/(n+1)$.

46. Encontre uma forma fechada para a função exponencial generalizada para a seqüência $\{a_n\}$, em que

a) $a_n = (-2)^n$. **b)** $a_n = -1$.
c) $a_n = n$. **d)** $a_n = n(n-1)$.
e) $a_n = 1/((n+1)(n+2))$.

47. Encontre a seqüência a partir das funções exponenciais generalizadas abaixo.

a) $f(x) = e^{-x}$
b) $f(x) = 3x^{2x}$
c) $f(x) = e^{3x} - 3e^{2x}$
d) $f(x) = (1-x) + e^{-2x}$
e) $f(x) = e^{-2x} - (1/(1-x))$
f) $f(x) = e^{-3x} - (1+x) + (1/(1-2x))$
g) $f(x) = e^{x^2}$

48. Encontre a seqüência a partir das funções exponenciais generalizadas abaixo.

a) $f(x) = e^{3x}$ **b)** $f(x) = 2e^{-3x+1}$
c) $f(x) = e^{4x} + e^{-4x}$ **d)** $f(x) = (1+2x) + e^{3x}$
e) $f(x) = e^x - (1/(1+x))$ **f)** $f(x) = xe^x$
g) $f(x) = e^{x^3}$

49. Um sistema de código codifica mensagens usando cadeias octais de dígitos (base 8). Um código é considerado válido se e somente se contiver um número par de 7s.

a) Encontre uma relação de recorrência linear e heterogênea para o número de códigos válidos de extensão n. Quais são as condições iniciais?

b) Resolva essa relação de recorrência usando o Teorema 6 da Seção 7.2.

c) Resolva essa relação de recorrência usando as funções generalizadas.

***50.** Um sistema de código codifica mensagens usando cadeias de dígitos de base 4 (ou seja, com dígitos do conjunto $\{0, 1, 2, 3\}$). Um código é válido se e somente se contiver um número par de 0s e um número par de 1s. Considere a_n como o número de códigos válidos de extensão n. Além disso, considere b_n, c_n e d_n como o número de cadeias de dígitos de base 4 de extensão n com um número par de 0s e um número ímpar de 1s, com um número ímpar de 0s e um número par de 1s e

com um número ímpar de 0s e um número ímpar de 1s, respectivamente.

a) Mostre que $d_n = 4^n - a_n - b_n - c_n$. Use isso para mostrar que $a_{n+1} = 2a_n + b_n + c_n$, $b_{n+1} = b_n - c_n + 4^n$ e $c_{n+1} = c_n - b_n + 4^n$.

b) Quais são a_1, b_1, c_1 e d_1?

c) Use os itens (a) e (b) para encontrar a_3, b_3, c_3 e d_3.

d) Use as relações de recorrência do item (a), juntamente com as condições iniciais do item (b), para organizar três equações relacionadas com as funções generalizadas $A(x)$, $B(x)$ e $C(x)$ para as seqüências $\{a_n\}$, $\{b_n\}$ e $\{c_n\}$, respectivamente.

e) Resolva o sistema de equações do item (d) para encontrar fórmulas explícitas para $A(x)$, $B(x)$ e $C(x)$ e utilize-as para encontrar as fórmulas explícitas para a_n, b_n, c_n e d_n.

Funções generalizadas são úteis no estudo do número de diferentes tipos de partições de um número inteiro n. Uma **partição** de um número inteiro positivo é uma forma de escrever este inteiro como a soma de números inteiros positivos com repetição e a ordem não sendo relevante. Por exemplo, as partições de 5 (sem restrições) são $1+1+1+1+1$, $1+1+1+2$, $1+1+3$, $1+2+2$, $1+4$, $2+3$ e 5. Os exercícios 51 a 56 ilustram alguns desses usos.

51. Mostre que o coeficiente $p(n)$ de x^n na expansão de série de potência formal de $1/((1-x)(1-x^2)(1-x^3)\cdots)$ é igual ao número de partições de n.

52. Mostre que o coeficiente $p_o(n)$ de x^n na expansão de série de potência formal de $1/((1-x)(1-x^3)(1-x^5)\cdots)$ é igual ao número de partições de n em número inteiros ímpares, ou seja, o número de maneiras de escrever n como a soma de números inteiros ímpares, em que a ordem não é relevante e podem ocorrer repetições.

53. Mostre que o coeficiente $p_d(n)$ de x^n na expansão de série de potência formal de $(1+x)(1+x^2)(1+x^3)\cdots$ é igual ao número de partições de n em partes distintas, ou seja, o número de maneiras de escrever n como a soma de números inteiros positivos, em que a ordem não é relevante, mas não podem ocorrer repetições.

54. Encontre $p_o(n)$, o número de partições de n em partes ímpares com repetições, e $p_d(n)$, o número de partições de n em partes distintas, para $1 \le n \le 8$, escrevendo cada partição de cada número inteiro.

55. Mostre que, se n for um número inteiro positivo, então o número de partições de n em partes distintas é igual ao número de partições de n em partes ímpares com repetições; ou seja, $p_o(n) = p_d(n)$. [*Dica:* Mostre que as funções generalizadas para $p_o(n)$ e $p_d(n)$ são iguais.]

****56.** (*Requer cálculo*) Utilize a função generalizada de $p(n)$ para mostrar que $p(n) \le e^{C\sqrt{n}}$ para qualquer constante C. [Hardy e Ramanujan mostraram que $p(n) \sim e^{\pi\sqrt{2/3}\sqrt{n}}/(4\sqrt{3n})$, que significa que a razão de $p(n)$ e o lado direito tendem a 1, assim como n tende ao infinito.]

Suponha que X seja uma variável aleatória no espaço amostral S, tal que $X(s)$ seja um número inteiro não negativo para todo $s \in S$. A **função de probabilidade generalizada** para X é

$$G_X(x) = \sum_{k=0}^{\infty} p(X(s) = k)x^k.$$

57. (*Requer cálculo*) Mostre que, se G_X for a função de probabilidade generalizada para uma variável aleatória X, tal que $X(s)$ é um número inteiro não negativo para todo $s \in S$, então

a) $G_X(1) = 1$.

b) $E(X) = G_X'(1)$.

c) $V(X) = G_X''(1) + G_X'(1) - G_X'(1)^2$.

58. Considere X como a variável aleatória cujo valor é n se o primeiro sucesso ocorrer no n-ésimo teste, quando testes independentes de Bernoulli forem realizados, cada um com probabilidade de sucesso p.

a) Encontre uma fórmula fechada para a função de probabilidade generalizada G_X.

b) Encontre o valor esperado e a variância de X usando o Exercício 57 e a forma fechada encontrada no item (a).

59. Considere m como um número inteiro positivo. Considere X_m como a variável aleatória cujo valor é n se o m-ésimo sucesso ocorrer no $(n+m)$-ésimo teste quando testes independentes de Bernoulli forem realizados, cada um com probabilidade de sucesso p.

a) Usando o Exercício 28 dos Exercícios Complementares do Capítulo 6, mostre que a função de probabilidade generalizada G_{X_m} é dada por $G_{X_m}(x) = p^m/(1-qx)^m$, em que $q = 1-p$.

b) Encontre o valor esperado e a variância de X_m usando o Exercício 57 e a forma fechada do item (a).

60. Mostre que se X e Y forem variáveis independentes aleatórias no espaço amostral S, tal que $X(s)$ e $Y(s)$ sejam números inteiros não negativos para todo $s \in S$, então $G_{X+Y}(x) = G_X(x)G_Y(x)$.

7.5 Inclusão–Exclusão

Introdução

Uma classe de matemática discreta contém 30 mulheres e 50 veteranos. Quantos estudantes na sala são ou mulheres ou veteranos? Esta questão não pode ser respondida a menos que sejam fornecidas mais informações. Adicionando o número de mulheres na sala e o número de veteranos provavelmente não nos dá a resposta correta, porque as mulheres veteranas são contadas duas vezes. Essa observação mostra que o número de estudantes na sala que são ou veteranos ou mulheres é a soma do número de mulheres com o número de veteranos na sala menos o número de

mulheres veteranas. Uma técnica para resolver esses problemas de contagem foi introduzida na Seção 5.1. Nesta seção, vamos generalizar as idéias introduzidas naquela seção para resolver os problemas que necessitam que contemos o número de elementos na união de mais de dois conjuntos.

O Princípio da Inclusão–Exclusão

Quantos elementos existem na união de dois conjuntos finitos? Na Seção 2.2, mostramos que o número de elementos na união de dois conjuntos A e B é a soma do número de elementos nos conjuntos menos o número de elementos de sua intersecção, ou seja,

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

Como mostramos na Seção 5.1, a fórmula para o número de elementos na união de dois conjuntos é útil em problemas de contagem. Os exemplos 1 a 3 fornecem ilustrações adicionais para o uso dessa fórmula.

EXEMPLO 1 Em uma classe de matemática discreta, todo estudante é graduando em ciência da computação ou matemática, ou em ambos. O número de estudantes que é graduando em ciência da computação (possivelmente com matemática) é 25; o número de estudantes que é graduando em matemática (possivelmente com ciência da computação) é 13 e o número de graduandos em ciência da computação e em matemática é 8. Quantos estudantes há na sala?

Solução: Considere A como o conjunto de estudantes graduandos em ciência da computação e B como o conjunto de estudantes graduandos em matemática. Então $A \cap B$ é o conjunto de estudantes na sala que são graduandos em matemática e em ciência da computação. Como todo estudante na sala é graduando ou em ciência da computação ou em matemática (ou em ambos), temos que o número de estudantes na sala é $|A \cup B|$. Assim,

$$\begin{aligned} |A \cup B| &= |A| + |B| - |A \cap B| \\ &= 25 + 13 - 8 \\ &= 30. \end{aligned}$$

Portanto, há 30 estudantes na sala. Essa computação está ilustrada na Figura 1. ◀

EXEMPLO 2 Quantos números inteiros positivos não excedentes a 1 000 são divisíveis por 7 ou 11?

Solução: Considere A como o conjunto de números inteiros positivos não excedentes a 1 000 que são divisíveis por 7, e B como o conjunto de números inteiros positivos não excedentes a 1 000 que são divisíveis por 11. Então, $A \cup B$ é o conjunto dos inteiros não excedentes a 1 000 que são divisíveis por 7 ou 11, e $A \cap B$ é o conjunto dos inteiros não excedentes a 1 000 que são divisíveis por 7 e 11. A partir do Exemplo 2 da Seção 3.4, sabemos que entre os números inteiros positivos não excedentes a 1 000 há $\lfloor 1\,000/7 \rfloor$ inteiros divisíveis por 7 e $\lfloor 1\,000/11 \rfloor$ divisíveis por 11. Como 7 e 11 são números primos entre si, os inteiros divisíveis por 7 e por 11 são aqueles divisíveis por $7 \cdot 11$. Conseqüentemente, há $\lfloor 1\,000/(11 \cdot 7) \rfloor$ números inteiros positivos não excedentes a 1 000 que são divisíveis por 7 e 11. Temos que há

$$\begin{aligned} |A \cup B| &= |A| + |B| - |A \cap B| \\ &= \left\lfloor \frac{1\,000}{7} \right\rfloor + \left\lfloor \frac{1\,000}{11} \right\rfloor - \left\lfloor \frac{1\,000}{7 \cdot 11} \right\rfloor \\ &= 142 + 90 - 12 \\ &= 220 \end{aligned}$$

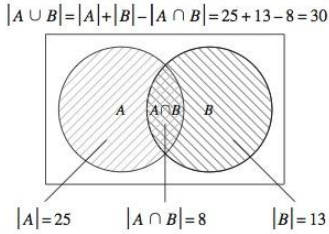


FIGURA 1 O Conjunto de Estudantes em uma Classe de Matemática Discreta.

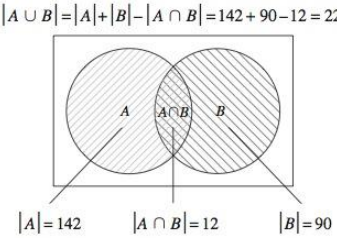


FIGURA 2 O Conjunto de Números Inteiros Positivos Não Excedentes a 1 000 Divisíveis por 7 ou 11.

números inteiros positivos não excedentes a 1 000 que são divisíveis por 7 ou por 11. Essa contagem está ilustrada na Figura 2. ◀

O Exemplo 3 mostra como encontrar o número de elementos em um conjunto universal finito que estão fora da união de dois conjuntos.

EXEMPLO 3 Suponha que existam 1 807 calouros em uma escola. Destes, 453 estão cursando ciência da computação, 567 cursam matemática e 299 estão cursando ambos, ciência da computação e matemática. Quantos calouros não estão cursando ciência da computação nem matemática?

Solução: Para encontrar o número de calouros que não cursam matemática ou ciência da computação, subtraia o número daqueles que estão nesses cursos do número total de calouros. Considere A como o conjunto de todos os calouros que estão cursando ciência da computação e B como o conjunto de todos os calouros que estão cursando matemática. Temos que $|A| = 453$, $|B| = 567$ e $|A \cap B| = 299$. O número de calouros que estão cursando ou ciência da computação ou matemática é

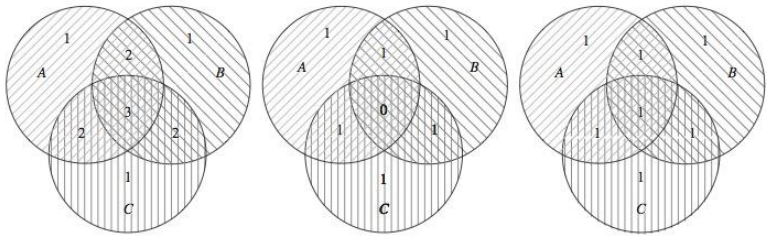
$$|A \cup B| = |A| + |B| - |A \cap B| = 453 + 567 - 299 = 721.$$

Conseqüentemente, há $1\,807 - 721 = 1\,086$ calouros que não estão cursando ciência da computação ou matemática. ◀

Começaremos agora o desenvolvimento de uma fórmula para o número de elementos na união de um número finito de conjuntos. A fórmula que será desenvolvida é chamada de **princípio da inclusão–exclusão.** Concretamente, antes de considerarmos as uniões de n conjuntos, em que n é qualquer número inteiro positivo, vamos derivar uma fórmula para o número de elementos na união de três conjuntos A, B e C. Para construir essa fórmula, notamos que $|A| + |B| + |C|$ conta cada elemento que está exatamente em um dos conjuntos uma vez, elementos que estão em dois conjuntos são contados duas vezes e elementos que estão nos três conjuntos, três vezes. Isso está ilustrado no primeiro painel da Figura 3.

Para retirar essa supercontagem dos elementos que estão em mais de um conjunto, subtraímos o número de elementos na intersecção de todos os pares dos três conjuntos. Obtemos

$$|A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C|.$$



(a) Contagem dos elementos por $|A|+|B|+|C|$

(b) Contagem dos elementos por $|A|+|B|+|C|-|A\cap B|-|A\cap C|-|B\cap C|$

(c) Contagem dos elementos por $|A|+|B|+|C|-|A\cap B|-|A\cap C|-|B\cap C|+|A\cap B\cap C|$

FIGURA 3 Encontrando uma Fórmula para o Número de Elementos na União dos Três Conjuntos.

Essa expressão ainda conta os elementos que aparecem em um conjunto uma vez. Um elemento que ocorre em dois conjuntos também é contado uma vez, pois esse elemento aparecerá em uma das três intersecções dos conjuntos. Entretanto, aqueles elementos que aparecem nos três conjuntos não serão contados por essa expressão, porque eles aparecem nas três intersecções dos conjuntos. Isso está ilustrado no segundo painel da Figura 3.

Para remediar esta não-contagem, adicionamos os números de elementos na intersecção dos três conjuntos. Essa expressão final conta cada elemento uma vez, sempre que ele estiver em um, dois ou três conjuntos. Assim,

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|.$$

Essa fórmula está ilustrada no terceiro painel da Figura 3.

O Exemplo 4 ilustra como essa fórmula pode ser usada.

EXEMPLO 4 Um total de 1 232 alunos cursam espanhol, 879 cursam francês e 114 cursam russo. Além disso, 103 cursam espanhol e francês, 23 cursam espanhol e russo e 14 cursam francês e russo. Se 2 092 estudantes cursam, pelo menos, umas das três línguas, espanhol, francês e russo, quantos estudantes cursam as três línguas?

Solução: Considere S como o conjunto de estudantes que cursam espanhol, F como o conjunto de estudantes que cursam francês e R como o conjunto de estudantes que cursam russo. Então,

$$|S| = 1\,232, \qquad |F| = 879, \qquad |R| = 114,$$

$$|S \cap F| = 103, \qquad |S \cap R| = 23, \qquad |F \cap R| = 14,$$

e

$$|S \cup F \cup R| = 2\,092.$$

Quando inserimos essas quantidades na equação

$$|S \cup F \cup R| = |S| + |F| + |R| - |S \cap F| - |S \cap R| - |F \cap R| + |S \cap F \cap R|$$

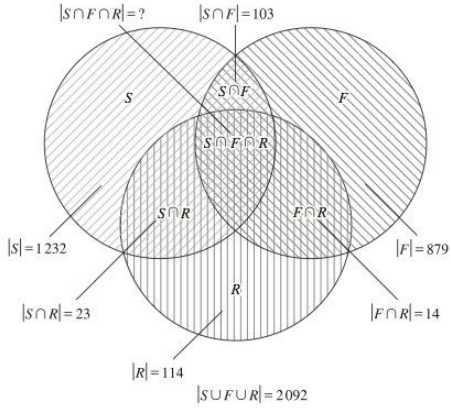


FIGURA 4 **O Conjunto de Estudantes que Cursam Espanhol, Francês e Russo.**

obtemos

$$2\,092 = 1\,232 + 879 + 114 - 103 - 23 - 14 + |S \cap F \cap R|.$$

Resolvemos agora em $|S \cap F \cap R|$. Encontramos que $|S \cap F \cap R| = 7$. Assim, há sete estudantes que cursam espanhol, francês e russo. Isso está ilustrado na Figura 4. ◀

Estabeleceremos e demonstraremos agora o princípio da inclusão–exclusão, que indica quantos elementos estão na união de um número finito de conjuntos finitos.

TEOREMA 1 **O PRINCÍPIO DA INCLUSÃO–EXCLUSÃO** Considere $A_1, A_2, \ldots, A_n$ como conjuntos finitos. Então,

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = \sum_{1 \le i \le n} |A_i| - \sum_{1 \le i < j \le n} |A_i \cap A_j| + \sum_{1 \le i < j < k \le n} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n|.$$

Demonstração: Vamos demonstrar a fórmula verificando que um elemento na união é contado exatamente uma vez pelo lado direito da equação. Suponha que a seja um membro de r dos conjuntos $A_1, A_2, \ldots, A_n$, em que $1 \le r \le n$. Este elemento é contado $C(r, 1)$ vezes por $\Sigma|A_i|$. É contado $C(r, 2)$ vezes por $\Sigma|A_i \cap A_j|$. Em geral, é contado $C(r, m)$ vezes pela somatória envolvendo m dos conjuntos A_i. Então, este elemento é contado exatamente

$$C(r, 1) - C(r, 2) + C(r, 3) - \cdots + (-1)^{r+1} C(r, r)$$

vezes pela expressão do lado direito dessa equação. Nosso objeto é avaliar essa quantidade. A partir do Corolário 2 da Seção 5.4, temos

$$C(r, 0) - C(r, 1) + C(r, 2) - \cdots + (-1)^r C(r, r) = 0.$$

Assim,

$$1 = C(r, 0) = C(r, 1) - C(r, 2) + \cdots + (-1)^{r+1} C(r, r).$$

Assim, cada elemento na união é contado exatamente uma vez pela expressão do lado direito da equação. Isso demonstra o princípio da inclusão–exclusão. ◁

O princípio da inclusão–exclusão fornece uma fórmula para o número de elementos na união de n conjuntos para cada número inteiro positivo n. Há termos nessa fórmula para o número de elementos na intersecção de todo subconjunto não vazio da coleção de n conjuntos. Assim, há $2^n - 1$ termos nessa fórmula.

EXEMPLO 5 Dê uma fórmula para o número de elementos na união de quatro conjuntos.

Exemplos Extras

Solução: O princípio da inclusão–exclusão mostra que

$$\begin{aligned} |A_1 \cup A_2 \cup A_3 \cup A_4| = {} & |A_1| + |A_2| + |A_3| + |A_4| \\ & - |A_1 \cap A_2| - |A_1 \cap A_3| - |A_1 \cap A_4| - |A_2 \cap A_3| - |A_2 \cap A_4| \\ & - |A_3 \cap A_4| + |A_1 \cap A_2 \cap A_3| + |A_1 \cap A_2 \cap A_4| + |A_1 \cap A_3 \cap A_4| \\ & + |A_2 \cap A_3 \cap A_4| - |A_1 \cap A_2 \cap A_3 \cap A_4|. \end{aligned}$$

Note que essa fórmula contém 15 termos diferentes, um para cada subconjunto não vazio de $\{A_1, A_2, A_3, A_4\}$. ◀

Exercícios

1. Quantos elementos estão em $A_1 \cup A_2$ se houver 12 elementos em A_1, 18 elementos em A_2 e
   a) $A_1 \cap A_2 = \emptyset$?
   b) $|A_1 \cap A_2| = 1$?
   c) $|A_1 \cap A_2| = 6$?
   d) $A_1 \subseteq A_2$?
2. Existem 345 estudantes em uma faculdade que cursam cálculo, 212 que cursam matemática discreta e 188 que cursam cálculo e matemática discreta. Quantos estudantes cursam ou cálculo ou matemática discreta?
3. Uma pesquisa em residências nos Estados Unidos revela que 96% têm, pelo menos, um aparelho de TV, 98% têm serviço de telefonia e 95% têm serviço de telefonia e, pelo menos, um aparelho de TV. Qual a porcentagem de residências nos Estados Unidos que não tem serviço de telefonia nem aparelho de TV?
4. Um relatório de marketing sobre computadores particulares afirma que 650.000 proprietários comprarão um modem para suas máquinas no próximo ano e 1.250.000 comprarão, pelo menos, um software. Se o relatório afirmar que 1.450.000 proprietários comprarão ou um modem ou, pelo menos, um software, quantos comprarão um modem e, pelo menos, um software?
5. Encontre o número de elementos em $A_1 \cup A_2 \cup A_3$ se houver 100 elementos em cada conjunto e se
   a) os conjuntos forem disjuntos dois a dois.
   b) houver 50 elementos comuns em cada par de conjuntos e não houver elementos nos três conjuntos simultaneamente.
   c) houver 50 elementos comuns em cada par de conjuntos e 25 elementos nos três conjuntos.
   d) os conjuntos forem iguais.
6. Encontre o número de elementos em $A_1 \cup A_2 \cup A_3$ se houver 100 elementos em A_1, 1.000 em A_2 e 10.000 em A_3, se
   a) $A_1 \subseteq A_2$ e $A_2 \subseteq A_3$.
   b) os conjuntos forem disjuntos dois a dois.
   c) houver dois elementos comuns em cada par de conjuntos e um elemento estiver nos três conjuntos.
7. Há 2 504 estudantes de ciência da computação em uma escola. Destes, 1 876 têm um curso sobre Pascal, 999 sobre Fortran e 345 sobre C. Além disso, 876 cursam as discipli-

nas sobre Pascal e Fortran, 231 sobre Fortran e C e 290 sobre Pascal e C. Se 189 desses estudantes cursam as disciplinas sobre Fortran, Pascal e C, quantos desses 2 504 estudantes não cursam nenhuma dessas três linguagens computacionais?

8. Em uma pesquisa com 270 universitários, encontrou-se que 64 gostam de couve-de-bruxelas, 94 de brócolis, 58 de couve-flor, 26 de couve-de-bruxelas e de brócolis, 28 de couve-de-bruxelas e de couve-flor, 22 de brócolis e de couve-flor e 14 dos três vegetais. Quantos destes 270 estudantes não gostam de nenhum desses vegetais?

9. Quantos estudantes estão matriculados em um curso de cálculo, matemática discreta, estruturas de dados ou linguagens de programação em uma faculdade, se há 507, 292, 312 e 344 estudantes nesses cursos, respectivamente; 14 fazem cálculo e estruturas de dados; 213, cálculo e linguagens de programação; 211, matemática discreta e estruturas de dados; 43, matemática discreta e linguagens de programação; e nenhum estudante faz cálculo e matemática discreta ou estruturas de dados e linguagens de programação concomitantemente?

10. Encontre o número de inteiros positivos não excedentes a 100 que não são divisíveis por 5 ou por 7.

11. Encontre o número de inteiros positivos não excedentes a 100 que são ou ímpares ou quadrados de um inteiro.

12. Encontre o número de inteiros positivos não excedentes a 1.000 que são ou quadrados ou cubos de um inteiro.

13. Quantas cadeias de bits de extensão oito não contêm seis 0s consecutivos?

***14.** Quantas permutações de 26 letras do alfabeto inglês não contêm qualquer uma das seqüências *fish, rat* ou *bird*?

15. Quantas permutações de 10 dígitos ou começam com os 3 dígitos 987, ou contêm os dígitos 45 na quinta e na sexta posições ou terminam com os 3 dígitos 123?

16. Quantos elementos fazem parte da união de quatro conjuntos se cada um dos conjuntos tiver 100 elementos, cada par desses conjuntos compartilharem 50 elementos, cada três desses conjuntos compartilharem 25 elementos e 5 elementos fizerem parte dos quatro conjuntos?

17. Quantos elementos fazem parte da união de quatro conjuntos se os conjuntos tiverem 50, 60, 70 e 80 elementos, respectivamente, cada par de conjuntos tiver 5 elementos em comum, cada trio de conjuntos tiver 1 elemento em comum e não houver nenhum elemento em comum nos quatro conjuntos?

18. Quantos termos existem na fórmula para o número de elementos na união de 10 conjuntos dada pelo princípio da inclusão–exclusão?

19. Desenvolva a fórmula explícita dada pelo princípio da inclusão–exclusão para o número de elementos na união de cinco conjuntos.

20. Quantos elementos há na união de cinco conjuntos se os conjuntos contêm 10 000 elementos cada, cada par de conjuntos tem 1 000 elementos em comum, cada trio de conjuntos tem 100 elementos em comum, cada quatro dos cinco conjuntos têm 10 elementos em comum e há 1 elemento em todos os cinco conjuntos?

21. Desenvolva a fórmula explícita dada pelo princípio da inclusão–exclusão para o número de elementos na união de seis conjuntos quando se sabe que três desses conjuntos não têm intersecção.

***22.** Demonstre o princípio da inclusão–exclusão usando a indução matemática.

23. Considere E_1, E_2 e E_3 como eventos do espaço amostral S. Encontre uma fórmula para a probabilidade de $E_1 \cup E_2 \cup E_3$.

24. Encontre a probabilidade de uma moeda não viciada ser lançada cinco vezes e aparecer três vezes coroa, no primeiro e no último lançamentos sair coroa ou no segundo e quarto lançamentos sair cara.

25. Encontre a probabilidade de escolher quatro números aleatoriamente de 1 a 100, inclusive, sem repetições, ou todos serem ímpares, ou divisíveis por 3, ou divisíveis por 5.

26. Encontre uma fórmula para a probabilidade da união de quatro eventos em um espaço amostral se três deles não podem ocorrer ao mesmo tempo.

27. Encontre uma fórmula para a probabilidade da união de cinco eventos em um espaço amostral se quatro deles não podem ocorrer ao mesmo tempo.

28. Encontre uma fórmula para a probabilidade da união de n eventos em um espaço amostral quando dois desses eventos não podem ocorrer ao mesmo tempo.

29. Encontre uma fórmula para a probabilidade da união de n eventos em um espaço amostral.

7.6 Aplicações da Inclusão–Exclusão

Introdução

Muitos problemas de contagem podem ser resolvidos usando o princípio da inclusão–exclusão. Por exemplo, podemos usar esse princípio para encontrar o número de primos menores que um número inteiro positivo. Muitos problemas podem ser resolvidos pela contagem do número de funções sobrejetoras de um conjunto finito a outro. O princípio da inclusão–exclusão pode ser utilizado para encontrar o número dessas funções. O famoso problema do chapéu pode ser resolvido usando o princípio da inclusão–exclusão. Esse problema procura a probabilidade de nenhuma pessoa pegar o chapéu correto de volta quando o guardador de chapéus os devolve aleatoriamente.

Uma Forma Alternativa de Inclusão–Exclusão

Há uma forma alternativa do princípio de inclusão–exclusão que é útil em problemas de contagem. Em particular, essa forma pode ser usada para resolver problemas que pedem pelo número de elementos em um conjunto que não tem nenhuma das n propriedades $P_1, P_2, \ldots, P_n$.

Considere A_i como o subconjunto que contém os elementos que têm propriedade P_i. O número de elementos com todas as propriedades, $P_{i_1}, P_{i_2}, \ldots, P_{i_k}$ será indicado por $N(P_{i_1} P_{i_2} \cdots P_{i_k})$. Escrevendo essas quantidades em termos de conjuntos, temos

$$|A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k}| = N(P_{i_1} P_{i_2} \cdots P_{i_k}).$$

Se o número de elementos com nenhuma das propriedades $P_1, P_2, \ldots, P_n$ for indicado por $N(P_1' P_2' \cdots P_n')$ e o número de elementos no conjunto for indicado por N, temos que

$$N(P_1' P_2' \cdots P_n') = N - |A_1 \cup A_2 \cup \cdots \cup A_n|.$$

A partir do princípio da inclusão–exclusão, vemos que

$$N(P'_1 P'_2 \cdots P'_n) = N - \sum_{1 \le i \le n} N(P_i) + \sum_{1 \le i < j \le n} N(P_i P_j) - \sum_{1 \le i < j < k \le n} N(P_i P_j P_k) + \cdots + (-1)^n N(P_1 P_2 \cdots P_n).$$

O Exemplo 1 mostra como o princípio da inclusão–exclusão pode determinar o número de soluções em números inteiros de uma equação com restrições.

EXEMPLO 1 Quantas soluções existem para

$$x_1 + x_2 + x_3 = 11,$$

em que x_1, x_2 e x_3 são números inteiros não negativos com $x_1 \le 3, x_2 \le 4$ e $x_3 \le 6$?

Solução: Para aplicar o princípio da inclusão–exclusão, considere uma solução que tenha a propriedade P_1 se $x_1 > 3$, a propriedade P_2 se $x_2 > 4$ e a propriedade P_3 se $x_3 > 6$. O número de soluções que satisfazem as inequações $x_1 \le 3, x_2 \le 4$ e $x_3 \le 6$ é

$$N(P_1' P_2' P_3') = N - N(P_1) - N(P_2) - N(P_3) + N(P_1 P_2) + N(P_1 P_3) + N(P_2 P_3) - N(P_1 P_2 P_3).$$

Usando as mesmas técnicas do Exemplo 5 da Seção 5.5, temos que

- N = número total de soluções $= C(3 + 11 - 1, 11) = 78$,
- $N(P_1)$ = (número de soluções com $x_1 \ge 4$) $= C(3 + 7 - 1, 7) = C(9, 7) = 36$,
- $N(P_2)$ = (número de soluções com $x_2 \ge 5$) $= C(3 + 6 - 1, 6) = C(8, 6) = 28$,
- $N(P_3)$ = (número de soluções com $x_3 \ge 7$) $= C(3 + 4 - 1, 4) = C(6, 4) = 15$,
- $N(P_1P_2)$ = (número de soluções com $x_1 \ge 4$ e $x_2 \ge 5$) $= C(3 + 2 - 1, 2) = C(4, 2) = 6$,
- $N(P_1P_3)$ = (número de soluções com $x_1 \ge 4$ e $x_3 \ge 7$) $= C(3 + 0 - 1, 0) = 1$,
- $N(P_2P_3)$ = (número de soluções com $x_2 \ge 5$ e $x_3 \ge 7$) $= 0$,
- $N(P_1P_2P_3)$ = (número de soluções com $x_1 \ge 4, x_2 \ge 5$ e $x_3 \ge 7$) $= 0$.

Inserindo essas quantidades na fórmula para $N(P_1'P_2'P_3')$, mostramos que o número de soluções com $x_1 \leq 3, x_2 \leq 4$ e $x_3 \leq 6$ é igual a

$$N(P_1'P_2'P_3') = 78 - 36 - 28 - 15 + 6 + 1 + 0 - 0 = 6. \quad \blacktriangleleft$$

O Crivo de Eratóstenes

O princípio da inclusão–exclusão pode ser utilizado para encontrar o número de primos não excedentes a determinado número inteiro positivo. Lembre-se de que um número inteiro composto é divisível por um primo não excedente à sua raiz quadrada. Assim, para encontrar o número de primos não excedentes a 100, primeiro note que os inteiros compostos não excedentes a 100 devem ter um fator primo não excedente a 10. Como os únicos primos menores que 10 são 2, 3, 5 e 7, os primos não excedentes a 100 são estes quatro primos e os números inteiros positivos maiores que 1 e não excedentes a 100 que não são divisíveis por 2, 3, 5 ou 7. Para aplicar o princípio da inclusão–exclusão, considere P_1 como a propriedade de um inteiro ser divisível por 2, considere P_2 como a propriedade de um inteiro ser divisível por 3, considere P_3 como a propriedade de um inteiro ser divisível por 5 e considere P_4 como a propriedade de um inteiro ser divisível por 7. Assim, o número de primos não excedentes a 100 é

$$4 + N(P_1'P_2'P_3'P_4').$$

Como há 99 números inteiros positivos maiores que 1 e não excedentes a 100, o princípio da inclusão–exclusão mostra que

$$\begin{aligned} N(P_1'P_2'P_3'P_4') = 99 &- N(P_1) - N(P_2) - N(P_3) - N(P_4) \\ &+ N(P_1P_2) + N(P_1P_3) + N(P_1P_4) + N(P_2P_3) + N(P_2P_4) + N(P_3P_4) \\ &- N(P_1P_2P_3) - N(P_1P_2P_4) - N(P_1P_3P_4) - N(P_2P_3P_4) \\ &+ N(P_1P_2P_3P_4). \end{aligned}$$

O número de inteiros não excedentes a 100 (e maiores que 1) que são divisíveis por todos os primos em um subconjunto de $\{2, 3, 5, 7\}$ é $\lfloor 100/N \rfloor$, em que N é o produto dos números primos nesse subconjunto. (Isto porque sempre dois desses primos não têm fator comum.) Conseqüentemente,

$$\begin{aligned} N(P_1'P_2'P_3'P_4') &= 99 - \left\lfloor \frac{100}{2} \right\rfloor - \left\lfloor \frac{100}{3} \right\rfloor - \left\lfloor \frac{100}{5} \right\rfloor - \left\lfloor \frac{100}{7} \right\rfloor \\ &\quad + \left\lfloor \frac{100}{2 \cdot 3} \right\rfloor + \left\lfloor \frac{100}{2 \cdot 5} \right\rfloor + \left\lfloor \frac{100}{2 \cdot 7} \right\rfloor + \left\lfloor \frac{100}{3 \cdot 5} \right\rfloor + \left\lfloor \frac{100}{3 \cdot 7} \right\rfloor + \left\lfloor \frac{100}{5 \cdot 7} \right\rfloor \\ &\quad - \left\lfloor \frac{100}{2 \cdot 3 \cdot 5} \right\rfloor - \left\lfloor \frac{100}{2 \cdot 3 \cdot 7} \right\rfloor - \left\lfloor \frac{100}{2 \cdot 5 \cdot 7} \right\rfloor - \left\lfloor \frac{100}{3 \cdot 5 \cdot 7} \right\rfloor + \left\lfloor \frac{100}{2 \cdot 3 \cdot 5 \cdot 7} \right\rfloor \\ &= 99 - 50 - 33 - 20 - 14 + 16 + 10 + 7 + 6 + 4 + 2 - 3 - 2 - 1 - 0 + 0 \\ &= 21. \end{aligned}$$

Assim, há $4 + 21 = 25$ números primos não excedentes a 100.

Links O **Crivo de Eratóstenes** é utilizado para encontrar todos os números primos não excedentes a determinado inteiro positivo. Por exemplo, o procedimento a seguir é utilizado para encontrar números primos não excedentes a 100. Primeiro, os números inteiros que são divisíveis por 2, diferentes de 2, são deletados. Como 3 é o primeiro número inteiro maior que 2, todos aqueles

TABELA 1 O Crivo de Eratóstenes.

Inteiros divisíveis por 2, diferentes de 2, são sublinhados.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Inteiros divisíveis por 3, diferentes de 3, são sublinhados.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Inteiros divisíveis por 5, diferentes de 5, são sublinhados.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Inteiros divisíveis por 7, diferentes de 7, são sublinhados; inteiros cinza são primos.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

inteiros divisíveis por 3, diferentes de 3, são deletados. Como 5 é o próximo número inteiro depois de 3, aqueles inteiros divisíveis por 5, diferentes de 5, são deletados. O próximo número inteiro é o 7, assim aqueles inteiros divisíveis por 7, diferentes de 7, são deletados. Como todos os números inteiros compostos não excedentes a 100 são divisíveis por 2, 3, 5 ou 7, todos os inteiros restantes, exceto 1, são primos. Na Tabela 1, os painéis mostram aqueles inteiros deletados em cada etapa, em que cada inteiro divisível por 2, diferente de 2, está sublinhado no primeiro painel, cada inteiro divisível por 3, diferente de 3, está sublinhado no segundo painel, cada inteiro divisível por 5, diferente de 5, está sublinhado no terceiro painel e cada inteiro divisível por 7,

Links

ERATÓSTENES (276–194 a.C.) Sabe-se que Eratóstenes nasceu em Cirene, uma colônia grega a oeste do Egito, e passou tempo estudando na Academia de Platão, em Atenas. Sabemos também que o Rei Ptolomeu II convidou Eratóstenes para ir à Alexandria como tutor de seu filho e que, logo depois, Eratóstenes se tornou chefe dos bibliotecários da famosa Biblioteca de Alexandria, um repositório central da sabedoria da Antiguidade. Eratóstenes era um acadêmico extremamente versátil, que escrevia sobre matemática, geografia, astronomia, história, filosofia e crítica literária. Além de seu estudo sobre matemática, ele é muito conhecido por sua cronologia da história antiga e por sua famosa medida da cicunferência da Terra.

diferente de 7, está sublinhado no quarto painel. Os inteiros que não estão sublinhados são os primos não excedentes a 100.

O Número de Funções Sobrejetivas

O princípio da inclusão–exclusão pode também ser utilizado para determinar o número de funções sobrejetivas de um conjunto com m elementos para um conjunto com n elementos. Primeiro, considere o Exemplo 2.

EXEMPLO 2 Quantas funções sobrejetivas existem de um conjunto com seis elementos para um conjunto com três elementos?

Solução: Suponha que os elementos no contradomínio sejam b_1, b_2 e b_3. Considere P_1, P_2 e P_3 como as propriedades de b_1, b_2 e b_3 não estarem na imagem da função, respectivamente. Note que uma função é sobrejetiva se e somente se não tiver nenhuma das propriedades P_1, P_2 ou P_3. Pelo princípio da inclusão–exclusão, temos que o número de funções sobrejetivas de um conjunto com seis elementos para um conjunto com três elementos é

$$N(P_1'P_2'P_3') = N - [N(P_1) + N(P_2) + N(P_3)] + [N(P_1P_2) + N(P_1P_3) + N(P_2P_3)] - N(P_1P_2P_3),$$

em que N é o número total de funções de um conjunto com seis elementos para um com três elementos. Vamos avaliar cada um dos termos do lado direito dessa equação.

A partir do Exemplo 6 da Seção 5.1, temos que $N = 3^6$. Note que $N(P_i)$ é o número de funções que não tem b_i em sua imagem. Assim, há duas opções para o valor da função em cada elemento do domínio. Portanto, $N(P_i) = 2^6$. Além disso, há $C(3, 1)$ termos desse tipo. Note que $N(P_i P_j)$ é o número de funções que não tem b_i e b_j em sua imagem. Assim, há apenas uma escolha para este valor da função em cada elemento do domínio. Portanto, $N(P_i P_j) = 1^6 = 1$. Além disso, há $C(3, 2)$ termos desse tipo. Note também que $N(P_1P_2P_3) = 0$, pois esse termo é o número de funções que não tem b_1, b_2 e b_3 em sua imagem. Claramente, não há essas funções. Portanto, o número de funções sobrejetivas de um conjunto de seis elementos para um com três elementos é

$$3^6 - C(3, 1)2^6 + C(3, 2)1^6 = 729 - 192 + 3 = 540. \quad \blacktriangleleft$$

O resultado geral que nos diz quantas funções sobrejetivas existem de um conjunto com m elementos para um com n elementos será agora estabelecido. A demonstração desse resultado foi deixada como exercício para o leitor.

TEOREMA 1 Considere m e n como números inteiros positivos, com $m \geq n$. Então, há

$$n^m - C(n, 1)(n-1)^m + C(n, 2)(n-2)^m - \cdots + (-1)^{n-1}C(n, n-1) \cdot 1^m$$

funções sobrejetivas de um conjunto com m elementos para um conjunto com n elementos.

Uma função sobrejetiva de um conjunto com m elementos para um conjunto com n elementos corresponde a uma maneira de distribuir os m elementos no domínio em n caixas idênticas, para que nenhuma caixa fique vazia, e então associar cada um dos n elementos do contradomínio a uma caixa. Isso significa que o número de funções sobrejetivas de um conjunto com m elementos para um conjunto com n elementos é o número de maneiras de distribuir m objetos diferentes em n caixas idênticas para que nenhuma caixa fique vazia, multiplicado pelo número de permutações de um conjunto com n elementos. Conseqüentemente, o número de funções sobrejetivas

de um conjunto com m elementos para um conjunto com n elementos é igual a $n!S(m, n)$, em que $S(m, n)$ é um *número de Stirling de segundo grau* definido na Seção 5.5. Isso significa que podemos utilizar o Teorema 1 para deduzir a fórmula dada na Seção 5.5 para $S(m, n)$. (Veja o Capítulo 6 de [MiRo91] para mais detalhes sobre números de Stirling de segundo grau.)

Uma das muitas aplicações diferentes do Teorema 1 será agora descrita.

EXEMPLO 3 Quantas maneiras há de distribuir cinco trabalhos diferentes para quatro funcionários diferentes se todo funcionário receber, pelo menos, um trabalho?

Solução: Considere a distribuição de trabalhos como uma função de um conjunto com cinco trabalhos para o conjunto de quatro empregados. Uma distribuição em que todo empregado recebe, pelo menos, um trabalho é o mesmo que uma função sobrejetiva de um conjunto de trabalhos para um conjunto de empregados. Assim, pelo Teorema 1, temos que há

$$4^5 - C(4, 1)3^5 + C(4, 2)2^5 - C(4, 3)1^5 = 1\,024 - 972 + 192 - 4 = 240$$

maneiras de distribuir os trabalhos para que cada empregado receba, pelo menos, um. ◀

Permutações Caóticas

O princípio da inclusão–exclusão será utilizado para resolver problemas de permutações de n objetos, em que nenhum objeto permanece em sua posição original. Considere o Exemplo 4.

EXEMPLO 4 O **Problema do Chapéu** Um funcionário novo recolhe os chapéus de n pessoas em um restaurante, esquecendo de anotar os números nos chapéus. Quando os clientes voltam para pegar seus chapéus, o funcionário devolve os chapéus, escolhendo-os aleatoriamente. Qual é a probabilidade de ninguém receber o chapéu correto? ◀

Lembre-se: A resposta é o número de maneiras como os chapéus podem ser organizados para que não haja nenhum em sua posição original dividido por $n!$, o número de permutações de n chapéus. Vamos retornar a este exercício depois que encontrarmos o número de permutações de n objetos, em que nenhum objeto permanece em sua posição original.

Uma **permutação caótica** é uma permutação de objetos na qual nenhum objeto permanece em sua posição original. Para resolver o problema proposto no Exemplo 4, precisaremos determinar o número de permutações caóticas de um conjunto com n objetos.

EXEMPLO 5 A permutação 21453 é uma permutação caótica de 12345, porque nenhum número foi deixado em sua posição original. Entretanto, 21543 não é uma permutação caótica de 12345, porque essa permutação mantém o 4 fixo. ◀

Considere D_n como o número de permutações caóticas de n objetos. Por exemplo, $D_3 = 2$, pois as permutações caóticas de 123 são 231 e 312. Avaliaremos D_n, para todos os números inteiros positivos n, usando o princípio da inclusão–exclusão.

TEOREMA 2 O número de permutações caóticas de um conjunto com n elementos é

$$D_n = n!\left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!}\right].$$

Demonstração: Considere uma permutação que tenha a propriedade P_i se fixar o elemento i. O número de permutações caóticas é o número de permutações que não tem a propriedade P_i para $i = 1, 2, \ldots, n$. Isso significa que

$$D_n = N(P_1' P_2' \cdots P_n').$$

Usando o princípio da inclusão–exclusão, temos que

$$D_n = N - \sum_i N(P_i) + \sum_{i<j} N(P_i P_j) - \sum_{i<j<k} N(P_i P_j P_k) + \cdots + (-1)^n N(P_1 P_2 \cdots P_n),$$

em que N é o número de permutações de n elementos. Essa equação afirma que o número de permutações que não fixam elementos é igual ao número total de permutações, menos o número das que fixam, pelo menos, um elemento, mais o número das que fixam, pelos menos, dois elementos, menos o número das que fixam, pelo menos, três elementos, e assim por diante. Todas as quantidades que aparecem do lado direito dessa equação serão agora encontradas.

Primeiro, note que $N = n!$, pois N é simplesmente o número total de permutações de n elementos. Também, $N(P_i) = (n-1)!$. Isso é obtido a partir da regra do produto, porque $N(P_i)$ é o número de permutações que fixam o elemento i, para que a i-ésima posição da permutação seja determinada, mas cada uma das posições restantes pode ser preenchida arbitrariamente. Da mesma forma,

$$N(P_i P_j) = (n-2)!,$$

pois este é o número de permutações que fixam os elementos i e j, mas os outros $n-2$ elementos podem ser arranjados arbitrariamente. Em geral, note que

$$N(P_{i_1} P_{i_2} \cdots P_{i_m}) = (n-m)!,$$

pois este é o número de permutações que fixam os elementos $i_1, i_2, \ldots, i_m$, mas os outros $n-m$ elementos podem ser arranjados arbitrariamente. Como há $C(n, m)$ maneiras de escolher m elementos a partir de n, temos que

$$\sum_{1 \le i \le n} N(P_i) = C(n,1)(n-1)!,$$
$$\sum_{1 \le i < j \le n} N(P_i P_j) = C(n,2)(n-2)!,$$

e em geral,

$$\sum_{1 \le i_1 < i_2 < \cdots < i_m \le n} N(P_{i_1} P_{i_2} \cdots P_{i_m}) = C(n,m)(n-m)!.$$

Links

NOTA HISTÓRICA Em *rencontres* (combinações), um antigo jogo de cartas francês, as 52 cartas do baralho são colocadas em uma fila. As cartas de um segundo baralho são colocadas acima de cada carta do primeiro baralho. O resultado é determinando contando o número de cartas que combinam nos dois baralhos. Em 1708, Pierre Raymond de Montmort (1678–1719) propôs *le problème de rencontres:* Qual a probabilidade de nenhuma combinação do jogo ocorrer? A solução para o problema de Montmort é a probabilidade de uma permutação selecionada aleatoriamente de 52 objetos ser uma permutação caótica, ou seja, $D_{52}/52!$, que, como veremos, é aproximadamente $1/e$.

TABELA 2 A Probabilidade de uma Permutação Caótica.						
n	2	3	4	5	6	7
$D_n/n!$	0,50000	0,33333	0,37500	0,36667	0,36806	0,36786

Conseqüentemente, inserindo essas quantidades em nossa fórmula para D_n, temos

$$\begin{aligned} D_n &= n! - C(n,1)(n-1)! + C(n,2)(n-2)! - \cdots + (-1)^n C(n,n)(n-n)! \\ &= n! - \frac{n!}{1!(n-1)!}(n-1)! + \frac{n!}{2!(n-2)!}(n-2)! - \cdots + (-1)^n \frac{n!}{n!0!}0!. \end{aligned}$$

Simplificando esta expressão, temos

$$D_n = n!\left[1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!}\right]. \quad \triangleleft$$

Agora, está simples encontrar D_n para um dado número inteiro positivo n. Por exemplo, usando o Teorema 2, temos que

$$D_3 = 3!\left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!}\right] = 6\left(1 - 1 + \frac{1}{2} - \frac{1}{6}\right) = 2,$$

como advertimos anteriormente.

Agora, a solução para o problema do Exemplo 4 pode ser dada.

Solução: A probabilidade de ninguém receber o chapéu correto é $D_n/n!$. A partir do Teorema 2, essa probabilidade é

$$\frac{D_n}{n!} = 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!}.$$

Os valores dessa probabilidade para $2 \leq n \leq 7$ estão na Tabela 2.

Usando os métodos de cálculo, podemos mostrar que

$$e^{-1} = 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!} + \cdots \sim 0{,}368.$$

Como esta é uma série alternante com termos que tendem a zero, temos que, quando n cresce ilimitadamente, a probabilidade de ninguém receber o chapéu correto converge para $e^{-1} \sim 0{,}368$. De fato, pode ser demonstrado que essa probabilidade aproxima $1/(n+1)!$ por e^{-1}. ◀

Exercícios

1. Suponha que em um caixote com 100 maçãs há 20 que contêm vermes e 15 que têm machucados. Apenas aquelas maçãs sem vermes ou machucados podem ser vendidas. Se houver 10 maçãs machucadas que contenham vermes, quantas das 100 maçãs podem ser vendidas?
2. Dos 1.000 inscritos para uma viagem de escalada no Himalaia, 450 têm problemas com altitude, 622 não estão em boa forma e 30 têm alergias. Um inscrito é qualificado se e somente se não tiver problema com altitude, estiver em boa forma e não tiver alergias. Se 111 inscritos têm problemas de altitude e não estão em boa forma, 14 têm problemas com altitude e alergia, 18 não estão em boa forma e têm alergia e 9 têm problema de altitude, não estão em boa forma e têm alergias, quantos inscritos estão qualificados?

3. Quantas soluções tem a equação $x_1 + x_2 + x_3 = 13$, em que x_1, x_2 e x_3 são números inteiros não negativos menores que 6?
4. Encontre o número de soluções para a equação $x_1 + x_2 + x_3 + x_4 = 17$, em que x_i, $i = 1, 2, 3, 4$, são números inteiros não negativos, tal que $x_1 \leq 3$, $x_2 \leq 4$, $x_3 \leq 5$ e $x_4 \leq 8$.
5. Encontre o número de primos menores que 200 usando o princípio da inclusão–exclusão.
6. Um número inteiro é chamado de **sem fator quadrático** se não for divisível pelo quadrado de um número inteiro positivo maior que 1. Encontre o número de inteiros positivos sem fator quadrático menores que 100.
7. Quantos números inteiros positivos menores que 10.000 não são a segunda potência, ou maior, de um inteiro?
8. Quantas funções sobrejetivas existem de um conjunto com sete elementos para um com cinco elementos?
9. Quantas maneiras há de distribuir seis brinquedos diferentes a três crianças diferentes, para que cada criança receba, pelo menos, um brinquedo?
10. De quantas maneiras oito bolas distintas podem ser distribuídas em três urnas distintas se cada urna deve conter, pelo menos, uma bola?
11. De quantas maneiras sete trabalhos diferentes podem ser distribuídos a quatro funcionários diferentes se cada empregado deve receber, pelo menos, um trabalho e o trabalho mais difícil deve ser designado ao melhor funcionário?
12. Liste todas as permutações caóticas de $\{1, 2, 3, 4\}$.
13. Quantas permutações caóticas há em um conjunto com sete elementos?
14. Qual é a probabilidade de nenhuma das 10 pessoas receberem o chapéu correto se o funcionário devolve os chapéus aleatoriamente?
15. Uma máquina que coloca cartas em envelopes passa a inserir as cartas aleatoriamente nos envelopes. Qual é a probabilidade de, em um grupo de 100 cartas,
    a) nenhuma carta ser colocada no envelope correto?
    b) exatamente uma carta ser colocada no envelope correto?
    c) exatamente 98 cartas serem colocadas nos envelopes corretos?
    d) exatamente 99 cartas serem colocadas nos envelopes corretos?
    e) todas as cartas serem colocadas nos envelopes corretos?
16. A um grupo de n estudantes são determinados assentos para cada uma das duas palestras na mesma sala. De quantas maneiras esses assentos podem ser designados se a nenhum estudante é designado o mesmo assento para ambas as palestras?

*17. De quantas maneiras os dígitos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 podem ser organizados para que nenhum dígito par permaneça em sua posição original?

*18. Use um argumento combinatório para mostrar que a seqüência $\{D_n\}$, em que D_n é o número de permutações caóticas de n objetos, satisfaz a relação de recorrência

$$D_n = (n - 1)(D_{n-1} + D_{n-2})$$

para $n \geq 2$. [*Dica:* Note que há $n - 1$ escolhas para o primeiro elemento k de uma permutação caótica. Considere separadamente as permutações caóticas que começam com k que tem ou não 1 na k-ésima posição.]

*19. Utilize o Exercício 18 para mostrar que

$$D_n = nD_{n-1} + (-1)^n$$

para $n \geq 1$.

20. Utilize o Exercício 19 para encontrar uma fórmula explícita para D_n.
21. Para quais números inteiros positivos n, D_n é o número de permutações caóticas de n objetos?
22. Suponha que p e q sejam números primos distintos. Utilize o princípio da inclusão–exclusão para encontrar $\phi(pq)$, o número de inteiros positivos não excedentes a pq que são primos relativos de pq.

*23. Utilize o princípio da inclusão–exclusão para obter uma fórmula para $\phi(n)$ quando a fatoração de n é

$$n = p_1^{a_1} p_2^{a_2} \cdots p_m^{a_m}.$$

*24. Mostre que se, n for um número inteiro positivo, então

$$n! = C(n, 0)D_n + C(n, 1)D_{n-1} + \cdots + C(n, n-1)D_1 + C(n, n)D_0,$$

em que D_k é o número de permutações caóticas de k objetos.

25. Quantas permutações caóticas de $\{1, 2, 3, 4, 5, 6\}$ começam com os números inteiros 1, 2 e 3, em qualquer ordem?
26. Quantas permutações caóticas de $\{1, 2, 3, 4, 5, 6\}$ terminam com os números inteiros 1, 2 e 3, em qualquer ordem?
27. Demonstre o Teorema 1.

Termos-chave e Resultados

TERMOS

Relação de recorrência: uma fórmula que expressa os termos de uma seqüência, exceto para alguns termos iniciais, como uma função de um ou mais termos anteriores da seqüência.

Condições iniciais para uma relação de recorrência: os valores dos termos de uma seqüência que satisfaz a relação de recorrência antes que esta relação tenha efeito.

Relação de recorrência linear e homogênea com coeficientes constantes: uma relação de recorrência que expressa os termos de uma seqüência, exceto os termos iniciais, como uma combinação linear de termos anteriores.

Raízes características de uma relação de recorrência linear e homogênea com coeficientes constantes: as raízes de um poli-

nômio associado a uma relação de recorrência linear e homogênea com coeficientes constantes.

Relação de recorrência linear e heterogênea com coeficientes constantes: uma relação de recorrência que expressa os termos de uma seqüência, exceto para os termos iniciais, como uma combinação linear de termos anteriores mais uma função que não é igual a zero e que depende apenas do índice.

Algoritmo de divisão e resolução: um algoritmo que resolve um problema recursivamente dividindo-o em um número fixo de problemas menores do mesmo tipo.

Função generalizada de uma seqüência: a seqüência formal que tem o n-ésimo termo da seqüência como coeficiente de x^n.

Crivo de Eratóstenes: um procedimento para encontrar os números primos menores que um número inteiro positivo específico.

Permutação caótica: uma permutação de objetos tal que nenhum objeto permanece em sua posição original.

RESULTADOS

A fórmula para o número de elementos na união de dois conjuntos finitos:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

A fórmula para o número de elementos na união de três conjuntos finitos:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

O princípio da inclusão–exclusão:

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = \sum_{1 \le i \le n} |A_i| - \sum_{1 \le i < j \le n} |A_i \cap A_j| + \sum_{1 \le i < j < k \le n} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n|$$

O número de funções sobrejetivas de um conjunto com m elementos para um conjunto com n elementos:

$$n^m - C(n, 1)(n-1)^m + C(n, 2)(n-2)^m - \cdots + (-1)^{n-1} C(n, n-1) \cdot 1^m$$

O número de permutações caóticas de n objetos:

$$D_n = n! \left| 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!} \right|$$

Questões de Revisão

1. a) O que é uma relação de recorrência?

b) Encontre uma relação de recorrência para a quantia que haverá em uma conta depois de n anos em que foram depositados R$ 1.000.000,00 em uma poupança com juros de 9% ao ano.

2. Explique como os números de Fibonacci são utilizados para resolver o problema de Fibonacci sobre coelhos.

3. a) Encontre uma relação de recorrência para o número de passos necessários para resolver o quebra-cabeça Torre de Hanói.

b) Mostre como essa relação de recorrência pode ser resolvida usando iteração.

4. a) Explique como encontrar uma relação de recorrência para o número de cadeias binárias de extensão n que não contém dois 1s consecutivos.

b) Descreva outro problema de contagem que tem uma solução que satisfaz a mesma relação de recorrência.

5. Defina uma relação de recorrência linear e homogênea de grau k.

6. a) Explique como resolver uma relação de recorrência linear e homogênea de grau 2.

b) Resolva a relação de recorrência $a_n = 13a_{n-1} - 22a_{n-2}$ para $n \ge 2$ se $a_0 = 3$ e $a_1 = 15$.

c) Resolva a relação de recorrência $a_n = 14a_{n-1} - 49a_{n-2}$ para $n \ge 2$ se $a_0 = 3$ e $a_1 = 35$.

7. a) Explique como encontrar $f(b^k)$, em que k é um número inteiro positivo se $f(n)$ satisfizer a relação de recorrência de divisão e resolução $f(n) = a\,f(n/b) + g(n)$ sempre que b dividir o número inteiro positivo n.

b) Encontre $f(256)$ se $f(n) = 3f(n/4) + 5n/4$ e $f(1) = 7$.

8. a) Desenvolva uma relação de recorrência de divisão e resolução para o número de comparações usadas para encontrar um número em uma lista usando uma busca binária.

b) Dê a estimativa O grande para o número de comparações utilizadas por uma busca binária a partir da relação de recorrência de divisão e resolução dada no item (a) usando o Teorema 1 da Seção 7.3.

9. a) Dê uma fórmula para o número de elementos na união de três conjuntos.

b) Explique por que essa fórmula é válida.

c) Explique como usar a fórmula do item (a) para encontrar o número de inteiros não excedentes a 1 000 que são divisíveis por 6, 10 ou 15.

d) Explique como usar a fórmula do item (a) para encontrar o número de soluções com inteiros não negativos para a equação $x_1 + x_2 + x_3 + x_4 = 22$, com $x_1 < 8$, $x_2 < 6$ e $x_3 < 5$.

10. a) Dê uma fórmula para o número de elementos na união de quatro conjuntos e explique por que ela é válida.

b) Suponha que os conjuntos A_1, A_2, A_3 e A_4 contenham cada um 25 elementos, a intersecção de dois conjuntos quaisquer contenha 5 elementos, a intersecção de três destes conjuntos contenha 2 elementos e 1 elemento pertença a todos os quatro conjuntos. Quantos elementos há na união dos quatro conjuntos?

11. a) Descreva o princípio da inclusão–exclusão.
b) Desenvolva uma demonstração desse princípio.

12. Explique como o princípio da inclusão–exclusão pode ser utilizado para contar o número de funções sobrejetivas de um conjunto com m elementos para um conjunto com n elementos.

13. a) Como podemos contar o número de maneiras de determinar m trabalhos para n funcionários para que cada funcionário receba, pelo menos, um trabalho?
b) Quantas maneiras há de determinar sete trabalhos a três funcionários para que cada funcionário receba, pelo menos, um trabalho?

14. Explique como o princípio da inclusão–exclusão pode ser utilizado para contar o número de números primos não excedentes ao número inteiro positivo n.

15. a) Defina uma permutação caótica.
b) Por que contar o número de maneiras de um empregado devolver os chapéus a n pessoas, de modo que ninguém receba o chapéu correto, é o mesmo que contar o número de permutações caóticas de n objetos?
c) Explique como contar o número de permutações caóticas de n objetos.

Exercícios Complementares

1. Um grupo de 10 pessoas começa uma corrente de cartas, com cada pessoa mandando uma carta para quatro outras pessoas. Cada uma dessas pessoas manda a carta para quatro pessoas diferentes entre si e diferentes das 10 originais.
a) Encontre uma relação de recorrência para o número de cartas enviadas na n-ésima etapa da corrente de cartas, se nenhuma pessoa recebe mais de uma carta.
b) Quais são as condições iniciais para a relação de recorrência do item (a)?
c) Quantas cartas são enviadas na n-ésima etapa da corrente de cartas?

2. Um reator nuclear criou 18 gramas de determinado isótopo radioativo. A cada uma hora, diminui 1% da massa do isótopo radioativo.
a) Construa uma relação de recorrência para a massa desse isótopo depois de n horas.
b) Quais são as condições iniciais para a relação de recorrência do item (a)?
c) Resolva essa relação de recorrência.

3. A cada hora, o governo dos Estados Unidos imprime 10 000 cédulas de \$ 1 a mais que na hora anterior, 4 000 de \$ 5 a mais que na hora anterior, 3 000 de \$ 10 a mais que na hora anterior, 2 500 de \$ 20 a mais que na hora anterior, 1 000 de \$ 50 a mais que na hora anterior e o mesmo número de cédulas de \$ 100 da última hora. Na primeira hora, 1 000 de cada uma das cédulas foram produzidas.
a) Construa uma relação de recorrência para a quantia de dinheiro produzida na n-ésima hora.
b) Quais são as condições iniciais para a relação de recorrência no item (a)?
c) Resolva a relação de recorrência para a quantia produzida na n-ésima hora.
d) Construa uma relação de recorrência para a quantia total produzida nas primeiras n horas.
e) Resolva a relação de recorrência para a quantia total produzida nas primeiras n horas.

4. Suponha que a toda hora haja duas novas bactérias para cada uma que estava presente na hora anterior em uma colônia e que todas as bactérias morram depois de 2 horas de vida. A colônia começa com 100 bactérias novas.
a) Construa uma relação de recorrência para o número de bactérias presentes depois de n horas.
b) Qual é a solução para essa relação de recorrência?
c) Quando a colônia terá mais de 1 milhão de bactérias?

5. Mensagens são enviadas por meio de um canal de comunicação usando dois sinais diferentes. Um sinal requer 2 microssegundos para a transmissão e o outro, 3 microssegundos. Cada sinal de uma mensagem é seguido imediatamente pelo próximo sinal.
a) Encontre uma relação de recorrência para o número de sinais diferentes que podem ser enviados em n microssegundos.
b) Quais são as condições iniciais da relação de recorrência no item (a)?
c) Quantas mensagens diferentes podem ser enviadas em 12 microssegundos?

6. Uma pequena agência de correio tem apenas selos de 4, 6 e 10 centavos. Encontre uma relação de recorrência para o número de maneiras de formar uma postagem de n centavos com esses selos se a ordem em que são utilizados for relevante. Quais são as condições iniciais para essa relação de recorrência?

7. Quantas maneiras são possíveis para fazer as postagens abaixo usando as regras descritas no Exercício 6?
a) 12 centavos
b) 14 centavos
c) 18 centavos
d) 22 centavos

8. Encontre as soluções do sistema simultâneo de congruências

$$a_n = a_{n-1} + b_{n-1}$$
$$b_n = a_{n-1} - b_{n-1}$$

com $a_0 = 1$ e $b_0 = 2$.

9. Resolva a relação de recorrência $a_n = a_{n-1}^2/a_{n-2}$ se $a_0 = 1$ e $a_1 = 2$. [*Dica:* Calcule os logaritmos de ambos os lados para obter uma relação de recorrência para a seqüência $\log a_n$, $n = 0, 1, 2, \ldots$.]

***10.** Resolva a relação de recorrência $a_n = a_{n-1}^3 a_{n-2}^2$ se $a_0 = 2$ e $a_1 = 2$. (Veja a dica para o Exercício 9.)

11. Encontre a solução para a relação de recorrência $a_n = 3a_{n-1} - 3a_{n-2} + a_{n-3} + 1$ se $a_0 = 2$, $a_1 = 4$ e $a_2 = 8$.

12. Encontre a solução para a relação de recorrência $a_n = 3a_{n-1} - 3a_{n-2} + a_{n-3}$ se $a_0 = 2$, $a_1 = 2$ e $a_2 = 4$.

***13.** Suponha que, no Exemplo 4 da Seção 7.1, um par de coelhos foge da ilha depois de se reproduzir duas vezes. Encontre uma relação de recorrência para o número de coelhos na ilha no n-ésimo mês.

14. Encontre a solução para a relação de recorrência $f(n) = 3f(n/5) + 2n^4$, em que n é divisível por 5, para $n = 5^k$, em que k é um número inteiro positivo e $f(1) = 1$.

15. Faça a estimativa do tamanho de f no Exercício 14 se f for uma função crescente.

16. Encontre a relação de recorrência que descreve o número de comparações utilizadas pelo algoritmo a seguir: Encontre o maior e o segundo maior elemento de uma seqüência de n números recursivamente dividindo a seqüência em duas subseqüências com um número igual de termos, ou em que não haja mais termos em uma subseqüência do que na outra, em cada etapa. Pare quando as subseqüências com dois termos forem encontradas.

17. Faça a estimativa do número de comparações usadas pelo algoritmo descrito no Exercício 16.

Considere $\{a_n\}$ como uma seqüência de números reais. As **diferenças progressivas** dessa seqüência são definidas recursivamente como a seguir: A **primeira diferença** é $\Delta a_n = a_{n+1} - a_n$; a **($k$+1)-ésima diferença** $\Delta^{k+1}a_n$ é obtida a partir de $\Delta^k a_n$ por $\Delta^{k+1}a_n = \Delta^k a_{n+1} - \Delta^k a_n$.

18. Encontre Δa_n, em que

a) $a_n = 3$.

b) $a_n = 4n + 7$.

c) $a_n = n^2 + n + 1$.

19. Considere $a_n = 3n^3 + n + 2$. Encontre $\Delta^k a_n$, em que k é igual a

a) 2.

b) 3.

c) 4.

***20.** Suponha que $a_n = P(n)$, em que P é um polinômio de grau d. Demonstre que $\Delta^{d+1}a_n = 0$ para todos os números inteiros não negativos n.

21. Considere $\{a_n\}$ e $\{b_n\}$ como seqüências de números reais. Mostre que

$$\Delta(a_n b_n) = a_{n+1}(\Delta b_n) + b_n(\Delta a_n).$$

22. Mostre que, se $F(x)$ e $G(x)$ são funções generalizadas para as seqüências $\{a_k\}$ e $\{b_k\}$, respectivamente, e c e d são números reais, então $(cF + dG)(x)$ é a função generalizada para $\{ca_k + db_k\}$.

23. (*Requer cálculo*) Este exercício mostra como as funções generalizadas podem ser utilizadas para resolver a relação de recorrência $(n + 1)a_{n+1} = a_n + (1/n!)$ para $n \geq 0$ com condição inicial $a_0 = 1$.

a) Considere $G(x)$ como a função generalizada para $\{a_n\}$. Mostre que $G'(x) = G(x) + e^x$ e $G(0) = 1$.

b) Mostre a partir do item (a) que $(e^{-x}G(x))' = 1$ e conclua que $G(x) = xe^x + e^x$.

c) Use o item (b) para encontrar uma forma fechada para a_n.

24. Suponha que 14 estudantes recebam um A na primeira prova em matemática discreta e 18 recebam um A na segunda prova. Se 22 estudantes receberam um A ou na primeira ou na segunda prova, quantos estudantes receberam um A nas duas provas?

25. Há 323 fazendas em Monmouth County que tem, pelo menos, um dos três tipos de animais: cavalos, vacas ou ovelhas. Se 224 têm cavalos, 85 têm vacas, 57 têm ovelhas e 18 fazendas têm os três tipos de animais, quantas fazendas têm dois dos três tipos de animais?

26. A restauração de uma base de dados de registros de estudantes em uma universidade produz o seguinte dado: Há 2.175 estudantes na universidade, 1.675 não são calouros, 1.074 estudantes estão matriculados em cálculo, 444 estudantes estão matriculados em matemática discreta, 607 estudantes não são calouros e estão matriculados em cálculo, 350 estudantes estão matriculados em cálculo e em matemática discreta, 201 estudantes não são calouros e estão matriculados em matemática discreta e 143 estudantes não são calouros e estão matriculados em cálculo e matemática discreta. Todas as respostas da restauração podem ser corretas?

27. Os estudantes de uma faculdade de matemática em uma universidade são graduandos em uma ou mais das seguintes áreas: matemática aplicada (MA), matemática pura (MP), pesquisas operacionais (PO) e ciência da computação (CC). Quantos estudantes há nesta faculdade se há 23 estudantes em MA; 17 estudantes em MP; 44 em PO; 63 em CC; 5 em MA e MP; 8 em MA e CC; 4 em MA e PO; 6 em MP e CC; 5 em MP e PO; 14 em PO e CC; 2 em MP, PO e CC; 2 em MA, PO e CC; 1 em MP, MA e PO; 1 em MP, MA e CC; e 1 nas quatro áreas.

28. Quantos termos são necessários quando o princípio da inclusão–exclusão é utilizado para expressar o número de elementos na união de sete conjuntos se não mais que cinco desses conjuntos têm um elemento comum?

29. Quantas soluções com números inteiros positivos existem para a equação $x_1 + x_2 + x_3 = 20$, com $2 < x_1 < 6$, $6 < x_2 < 10$ e $0 < x_3 < 5$?

30. Quantos números inteiros positivos menores que 1 000 000

a) são divisíveis por 2, 3 ou 5?

b) não são divisíveis por 7, 11 ou 13?

c) são divisíveis por 3, mas não por 7?

31. Quantos números inteiros positivos menores que 200 são

a) a segunda potência, ou maior, de inteiros?

b) a segunda potência, ou maior, de inteiros ou primos?

c) não divisíveis pelo quadrado de um número inteiro maior que 1?

d) não divisíveis pelo cubo de um inteiro maior que 1?

e) não divisíveis por três ou mais primos?

***32.** Quantas maneiras há de designar seis trabalhos diferentes para três funcionários diferentes se o trabalho mais difícil

for designado ao funcionário mais experiente e o trabalho mais fácil, ao funcionário menos experiente?

33. Qual é a probabilidade de uma pessoa ter seu chapéu devolvido corretamente pelo guardador de chapéus que os devolve a n pessoas aleatoriamente?
34. Quantas cadeias de bits de extensão seis não contêm quatro 1s consecutivos?
35. Qual é a probabilidade de uma cadeia de bits de extensão seis conter, pelo menos, quatro 1s?

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dado um número inteiro positivo n, liste todos os movimentos necessários na Torre de Hanói para mover n discos de um pino a outro de acordo com as regras do jogo.
2. Dado um número inteiro positivo n e um número inteiro k com $1 \leq k \leq n$, liste todos os movimentos utilizados pelo algoritmo de Frame–Stewart (descrito no preâmbulo do Exercício 54 da Seção 7.1) para mover n discos de um pino a outro usando quatro pinos de acordo com as regras do jogo.
3. Dado um número inteiro positivo n, liste todas as cadeias de bits de extensão n que não têm um par consecutivo de 0s.
4. Dado um número inteiro positivo n, desenvolva todas as maneiras de colocar parênteses no produto de $n + 1$ variáveis.
5. Dada uma relação de recorrência $a_n = c_1 a_{n-1} + c_2 a_{n-2}$, em que c_1 e c_2 são números reais, com condições iniciais $a_0 = C_0$ e $a_1 = C_1$ e um número inteiro positivo k, encontre a_k usando iteração.
6. Dada uma relação de recorrência $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ e condições iniciais $a_0 = C_0$ e $a_1 = C_1$, determine a solução única dessa relação.
7. Dada uma relação de recorrência na forma $f(n) = a f(n/b) + c$, em que a é um número real, b é um número inteiro positivo e c é um número real, e um número inteiro positivo k, encontre $f(b^k)$ usando iteração.
8. Dado o número de elementos na intersecção de três conjuntos, o número de elementos em cada intersecção de um par de conjuntos e o número de elementos em cada conjunto, encontre o número de elementos em sua união.
9. Dado um número inteiro positivo n, produza a fórmula para o número de elementos na união de n conjuntos.
10. Dados os números inteiros positivos m e n, encontre o número de funções sobrejetivas de um conjunto com m elementos para um conjunto com n elementos.
11. Dado um número inteiro positivo n, liste todas as permutações caóticas do conjunto $\{1, 2, 3, \ldots, n\}$.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Encontre o valor de f_{100}, f_{500} e f_{1000}, em que f_n é o n-ésimo número de Fibonacci.
2. Encontre o menor número de Fibonacci maior que 1 000 000, maior que 1 000 000 000 e maior que 1 000 000 000 000.
3. Encontre quantos números primos de Fibonacci você puder. Não sabemos se eles são infinitos.
4. Descreva todos os movimentos necessários para resolver a Torre de Hanói com 10 discos.
5. Descreva todos os movimentos necessários para usar o algoritmo de Frame–Stewart para mover 20 discos de um pino ao outro usando quatro pinos, de acordo com as regras do *Reve's puzzle*.
6. Verifique a conjectura de Frame para resolver o *Reve's puzzle*, para n discos para tantos números inteiros n quantos forem possíveis, mostrando que o quebra-cabeça não pode ser resolvido usando menos movimentos do que aqueles feitos pelo algoritmo de Frame–Stewart com a escolha ótima de k.
7. Calcule o número de operações necessárias para multiplicar dois números inteiros com n bits por vários números inteiros n, incluindo 16, 64, 256 e 1.024, usando a multiplicação rápida descrita na Seção 7.3 e o algoritmo para multiplicação de inteiros (Algoritmo 3 da Seção 3.6).
8. Calcule o número de operações necessárias para multiplicar duas matrizes $n \times n$ para vários números inteiros n, incluindo 4, 16, 64 e 128, usando a multiplicação rápida de matrizes descrita na Seção 7.3 e o algoritmo para multiplicação de matrizes (Algoritmo 1 da Seção 3.8).
9. Utilize o crivo de Eratóstenes para encontrar todos os números primos não excedentes a 1.000.
10. Encontre o número de primos não excedentes a 10 000 usando o método descrito na Seção 7.6 para encontrar o número de primos não excedentes a 100.
11. Liste todas as permutações caóticas de $\{1, 2, 3, 4, 5, 6, 7, 8\}$.
12. Compute a probabilidade de uma permutação de n objetos ser uma permutação caótica para todos os números inteiros positivos não excedentes a 20 e determine o quão rápido essas probabilidades se aproximam do número e.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Encontre o recurso original em que Fibonacci apresentou seu desafio sobre a modelagem da população de coelhos. Discuta esse problema e outros propostos por Fibonacci e dê algumas informações sobre ele.
2. Explique como os números de Fibonacci aparecem em uma grande variedade de aplicações, tais como filotaxia, o estudo da harmonia no aparecimento de folhas nas plantas, no estudo de reflexão de espelhos, e assim por diante.
3. Descreva variações diferentes do quebra-cabeça Torre de Hanói, incluindo aquelas com mais de três pinos (incluindo o *Reve's puzzle* discutido no texto e nos exercícios), aquelas nas quais os movimentos dos discos são restritos e aquelas nas quais os discos têm o mesmo tamanho. Inclua o que se sabe sobre o número de movimentos necessários para resolver cada variante.
4. Discuta quantos problemas você conseguir em que aparecem os números de Catalan.
5. Descreva a solução para o problema de Ulam (veja o Exercício 28 da Seção 7.3) que envolve a busca com um truque encontrado por Andrzej Pelc.
6. Discuta as variações do problema de Ulam (veja o Exercício 28 da Seção 7.3) que envolvem a busca com mais de um truque e quanto se sabe sobre esse problema.
7. Defina o corpo convexo de um conjunto de pontos em um plano e descreva os algoritmos de divisão e resolução para encontrar o corpo convexo de um conjunto de pontos no plano.
8. Procure pela definição dos *lucky numbers*. Explique como eles são encontrados usando uma técnica semelhante ao crivo de Eratóstenes. Encontre todos os *lucky numbers* menores que 1.000.
9. Descreva como os métodos de seleção são utilizados na teoria dos números. Que tipos de resultados são estabelecidos usando esses métodos?
10. Procure pelas regras do jogo de cartas francês *rencontres*. Descreva essas regras e o trabalho de Pierre Raymond de Montmort em *le problème de rencontres*.
11. Descreva como as funções exponenciais generalizadas podem ser utilizadas para resolver uma variedade de problemas de contagem.
12. Descreva a teoria de Polyá sobre contagem e o tipo de problemas de contagem que podem ser resolvidos utilizando essa teoria.
13. O *problème des ménages* (o problema dos casais) pergunta sobre o número de maneiras de distribuir n casais em uma mesa para que sentem sexos alternados e não fiquem juntos marido e mulher. Explique o método usado por E. Lucas para resolver esse problema.
14. Explique como *rook polynomials* podem ser utilizados para resolver problemas de contagem.

CAPÍTULO

8 Relações

8.1 Relações e Suas Propriedades
8.2 Relações *n*-árias e Suas Aplicações
8.3 Representação de Relações
8.4 Fecho de Relações
8.5 Relações de Equivalência
8.6 Ordens Parciais

Ligações entre elementos de conjuntos ocorrem em muitos contextos. Todos os dias lidamos com ligações tais como aquelas entre um negócio e seu número de telefone, um empregado e seu salário, uma pessoa e um parente, e assim por diante. Em matemática, estudamos ligações como aquelas entre um número inteiro positivo e um que o divida, um inteiro e um que seja congruente ao módulo 5, um número real e um que seja maior do que ele, um número real x e o valor $f(x)$ quando f for uma função, e assim por diante. Ligações como aquela entre um programa e uma variável que ele use e como aquela entre uma linguagem computacional e uma sentença válida nessa linguagem ocorrem com freqüência na ciência da computação.

As ligações entre os elementos de conjuntos são representadas usando a estrutura chamada de relação, que é simplesmente um subconjunto do produto cartesiano dos conjuntos. Relações podem ser usadas para resolver problemas tais como a determinação de quais pares de cidades são ligados por linhas aéreas em uma rede, a busca de uma ordem viável para as diferentes fases de um projeto complicado ou na produção de uma maneira útil de armazenar informações em bancos de dados computacionais.

Em algumas linguagens computacionais, apenas os primeiros 31 caracteres do nome de uma variável importam. A relação que consiste em pares ordenados de seqüências de caracteres, em que a primeira seqüência tem os mesmos 31 caracteres iniciais que a segunda, é um exemplo de um tipo especial de relação, conhecida como uma relação de equivalência. As relações de equivalência aparecem em toda a matemática e em ciência da computação. Neste capítulo, estudaremos relações de equivalência e outros tipos especiais de relações.

8.1 Relações e Suas Propriedades

Introdução

A maneira mais direta de expressar relações entre elementos de dois conjuntos é usar pares ordenados compostos pelos dois elementos relacionados. Por esta razão, conjuntos de pares ordenados são chamados de relações binárias. Nesta seção, introduziremos a terminologia básica usada para descrever relações binárias. Mais adiante, neste capítulo, usaremos essas relações para resolver problemas com redes de comunicação, cronograma de projeto e identificação de elementos em conjuntos com propriedades comuns.

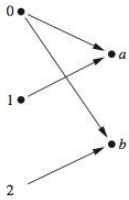
DEFINIÇÃO 1 Sejam A e B dois conjuntos. Uma *relação binária de A em B* é um subconjunto de $A \times B$.

Em outras palavras, uma relação binária de A em B é um conjunto R de pares ordenados em que o primeiro elemento de cada par ordenado vem de A e o segundo elemento vem de B. Usamos a notação $a\,R\,b$ para indicar que $(a, b) \in R$ e $a \not R\, b$ para indicar que $(a, b) \notin R$. Além disso, quando (a, b) pertence a R, diz-se que a está **relacionado** a b por R.

Relações binárias representam ligações entre elementos de dois conjuntos. Introduziremos relações n-árias, que expressam ligações entre elementos de mais de dois conjuntos, mais a frente, neste capítulo. Omitiremos a palavra *binária* quando não houver perigo de confusão.

Os exemplos 1 a 3 ilustram a noção de relação.

EXEMPLO 1 Seja A o conjunto de estudantes de sua escola e B, o conjunto de disciplinas. Seja R a relação que consiste naqueles pares (a, b), no qual a é um aluno matriculado na disciplina b. Por



R	a	b
0	×	×
1	×	
2		×

FIGURA 1 **Mostrando os Pares Ordenados na Relação R do Exemplo 3.**

exemplo, se Jason Goodfriend e Deborah Sherman estão matriculados em CS518, os pares (Jason Goodfriend, CS518) e (Deborah Sherman, CS518) pertencem a R. Se Jason Goodfriend também estiver matriculado em CS510, então o par (Jason Goodfriend, CS510) também está em R. Entretanto, se Deborah Sherman não estiver matriculada em CS510, então o par (Deborah Sherman, CS510) não está em R.

Observe que se um estudante, no momento, não estiver matriculado em nenhuma disciplina, não existirá nenhum par em R que tenha esse estudante como primeiro elemento. Analogamente, se uma disciplina não estiver sendo oferecida no momento, não existirá nenhum par em R que tenha essa disciplina como seu segundo elemento. ◀

EXEMPLO 2 Sejam A o conjunto do nome de todas as cidades e B o conjunto dos 50 estados dos Estados Unidos da América. Defina a relação R especificando que (a, b) pertence a R se a cidade a estiver no estado b. Por exemplo, (Boulder, Colorado), (Bangor, Maine), (Ann Arbor, Michigan), (Middletown, Nova Jersey), (Middletown, Nova York), (Cupertino, Califórnia) e (Red Bank, Nova Jersey) estão em R. ◀

EXEMPLO 3 Sejam $A = \{0, 1, 2\}$ e $B = \{a, b\}$. Então, $\{(0, a), (0, b), (1, a), (2, b)\}$ é uma relação de A para B. Isso significa, por exemplo, que $0\ R\ a$, mas que $1 \not R\ b$. As relações podem ser representadas graficamente, como mostrado na Figura 1, usando flechas para indicar pares ordenados. Uma outra maneira de representar essa relação é usar uma tabela, o que também é feito na Figura 1. Discutiremos representações de relações, com mais detalhes, na Seção 8.3. ◀

Funções como Relações

Lembre-se de que uma função f de um conjunto A para um conjunto B (conforme definido na Seção 2.3) associa exatamente um elemento de B a cada elemento de A. O gráfico de f é o conjunto de pares ordenados (a, b), tal que $b = f(a)$. Como o gráfico de f é um subconjunto de $A \times B$, ele é uma relação de A em B. Além disso, o gráfico de uma função tem a propriedade que cada elemento de A é o primeiro elemento de, exatamente, um par ordenado do gráfico.

Reciprocamente, se R for uma relação de A em B tal que todo elemento de A seja o primeiro elemento de, exatamente, um par ordenado de R, então é possível definir uma função que tenha R como seu gráfico. Isso pode ser feito associando a um elemento a de A o único elemento $b \in B$, tal que $(a, b) \in R$. (Observe que a relação R no Exemplo 2 não é o gráfico de uma função, pois Middletown ocorre mais de uma vez como primeiro elemento de um par ordenado em R.)

Uma relação pode ser usada para expressar ligações de um para muitos elementos entre os elementos dos conjuntos A e B (como no Exemplo 2), em que um elemento de A pode estar relacionado a mais de um elemento de B. Uma função representa uma relação em que exatamente um elemento de B está relacionado a cada elemento de A.

As relações são uma generalização de funções; elas podem ser usadas para expressar uma classe muito mais ampla de ligações entre conjuntos.

Relações em um Conjunto

Relações de um conjunto A em si próprio são de interesse especial.

DEFINIÇÃO 2 Uma *relação no conjunto A* é uma relação de A em A.

Em outras palavras, uma relação em um conjunto A é um subconjunto de $A \times A$.

EXEMPLO 4 Seja A o conjunto $\{1, 2, 3, 4\}$. Quais pares ordenados estão na relação $R = \{(a,b) \mid a \text{ divide } b\}$?

Solução: Como (a, b) está em R se e somente se a e b forem inteiros positivos que não excedam a 4, tal que a divide b, vemos que

$$R = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 4), (3, 3), (4, 4)\}.$$

Os pares nessa relação estão mostrados tanto graficamente como na forma tabular na Figura 2. ◀

A seguir, no Exemplo 5, serão dadas algumas relações no conjunto dos inteiros.

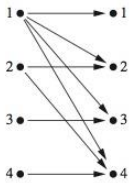
EXEMPLO 5 Considere estas relações no conjunto dos inteiros:

$$R_1 = \{(a, b) \mid a \leq b\},$$
$$R_2 = \{(a, b) \mid a > b\},$$
$$R_3 = \{(a, b) \mid a = b \text{ ou } a = -b\},$$
$$R_4 = \{(a, b) \mid a = b\},$$
$$R_5 = \{(a, b) \mid a = b + 1\},$$
$$R_6 = \{(a, b) \mid a + b \leq 3\}.$$

Quais dessas relações contêm cada um dos pares $(1, 1)$, $(1, 2)$, $(2, 1)$, $(1, -1)$ e $(2, 2)$?

Lembre-se: De modo diferente das relações dos exemplos 1 a 4, estas são relações em um conjunto infinito.

Solução: O par $(1, 1)$ está em R_1, R_3, R_4 e R_6; $(1, 2)$ está em R_1 e R_6; $(2, 1)$ está em R_2, R_5 e R_6; $(1, -1)$ está em R_2, R_3 e R_6; e, finalmente, $(2, 2)$ está em R_1, R_3 e R_4. ◀



R	1	2	3	4
1	×	×	×	×
2		×		×
3			×	
4				×

FIGURA 2 Mostrando os Pares Ordenados na Relação R do Exemplo 4.

Não é difícil determinar o número de relações em um conjunto finito, pois uma relação em um conjunto A é simplesmente um subconjunto de $A \times A$.

EXEMPLO 6 Quantas relações existem em um conjunto com n elementos?

Solução: Uma relação em um conjunto A é um subconjunto de $A \times A$. Como $A \times A$ tem n^2 elementos quando A tem n elementos e um conjunto com m elementos tem 2^m subconjuntos, existem 2^{n^2} subconjuntos de $A \times A$. Logo, existem 2^{n^2} relações em um conjunto com n elementos. Por exemplo, existem $2^{3^2} = 2^9 = 512$ relações no conjunto $\{a, b, c\}$. ◀

Propriedades de Relações

Existem diversas propriedades que são usadas para classificar relações em um conjunto. Introduziremos agora as mais importantes delas.

Em algumas relações, um elemento está sempre relacionado a si mesmo. Por exemplo, seja R a relação no conjunto de todas as pessoas que consiste nos pares (x, y), em que x e y tem a mesma mãe e o mesmo pai. Então $x\,R\,x$ para toda pessoa x.

DEFINIÇÃO 3 Uma relação R em um conjunto A é chamada de *reflexiva* se $(a, a) \in R$ para todo elemento $a \in A$.

Lembre-se: Usando quantificadores, vemos que a relação R no conjunto A é reflexiva se $\forall a((a, a) \in R)$, em que o universo do discurso é o conjunto de todos os elementos de A.

Vemos que uma relação em A é reflexiva se todo elemento de A está relacionado a si mesmo. Os exemplos 7 a 9 ilustram o conceito de uma relação reflexiva.

EXEMPLO 7 Considere as seguintes relações em $\{1, 2, 3, 4\}$:

$R_1 = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 4), (4, 1), (4, 4)\}$,
$R_2 = \{(1, 1), (1, 2), (2, 1)\}$,
$R_3 = \{(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (3, 3), (4, 1), (4, 4)\}$,
$R_4 = \{(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)\}$,
$R_5 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\}$,
$R_6 = \{(3, 4)\}$.

Quais dessas relações são reflexivas?

Solução: As relações R_3 e R_5 são reflexivas, pois ambas contêm todos os pares da forma (a, a), ou seja, $(1, 1)$, $(2, 2)$, $(3, 3)$ e $(4, 4)$. As outras relações não são reflexivas, pois elas não contêm todos esses pares ordenados. Em particular, R_1, R_2, R_4 e R_6 não são reflexivas, pois $(3, 3)$ não está em nenhuma dessas relações. ◀

EXEMPLO 8 Quais das relações do Exemplo 5 são reflexivas?

Solução: As relações reflexivas do Exemplo 5 são R_1 (pois $a \leq a$ para todo inteiro a), R_3 e R_4. Para cada uma das outras relações deste exemplo, é fácil encontrar um par da forma (a, a) que não esteja na relação. (Isto é deixado como exercício para o leitor.) ◀

EXEMPLO 9 A relação "divide", no conjunto dos inteiros positivos, é reflexiva?

Solução: Como $a \mid a$ sempre que a for um inteiro positivo, a relação "divide" é reflexiva. (Observe que se substituirmos o conjunto dos inteiros positivos pelo conjunto de todos os inteiros, a relação não será reflexiva, pois 0 não divide 0.) ◀

Em algumas relações, um elemento está relacionado a um segundo elemento se e somente se o segundo elemento também estiver relacionado ao primeiro. A relação que consiste nos pares (x, y), em que x e y são estudantes em sua escola que tem, pelo menos, uma aula em comum, tem esta propriedade. Outras relações têm a propriedade que se um elemento estiver relacionado a um segundo elemento, então este segundo elemento não estará relacionado ao primeiro. A relação que consiste nos pares (x, y), em que x e y são estudantes em sua escola, com x tendo uma média geral mais alta do que y, tem esta propriedade.

DEFINIÇÃO 4 Uma relação R em um conjunto A é chamada de *simétrica* se $(b, a) \in R$ sempre que $(a, b) \in R$, para quaisquer $a, b \in A$. Uma relação R em um conjunto A tal que para quaisquer $a, b \in A$, se $(a, b) \in R$ e $(b, a) \in R$, então $a = b$ é chamada de *anti-simétrica*.

Lembre-se: Usando quantificadores, vemos que a relação R em um conjunto A é simétrica se $\forall a \, \forall b \, ((a, b) \in R \rightarrow (b, a) \in R)$. Analogamente, a relação R em um conjunto A é anti-simétrica se $\forall a \, \forall b \, (((a, b) \in R \wedge (b, a) \in R) \rightarrow (a = b))$.

Isto é, uma relação é simétrica se e somente se a está relacionado a b implicar que b está relacionado a a. Uma relação é anti-simétrica se e somente se não existir nenhum par de elementos distintos a e b com a relacionado a b e b relacionado a a. Isto é, a única maneira de ter a relacionado a b e b relacionado a a é que a e b sejam o mesmo elemento. Os termos *simétrica* e *anti-simétrica* não são opostos, pois uma relação pode ter ambas as propriedades ou não ter nenhuma delas (ver o Exercício 8 no final desta seção). Uma relação não pode ser tanto simétrica quanto anti-simétrica se ela contiver algum par da forma (a, b), com $a \neq b$.

Lembre-se: Embora relativamente poucas das 2^{n^2} relações em um conjunto com n elementos sejam simétricas ou anti-simétricas, como um argumento de contagem pode mostrar, muitas relações importantes têm uma dessas propriedades. (Ver o Exercício 45.)

EXEMPLO 10 Quais das relações do Exemplo 7 são simétricas e quais são anti-simétricas?

Exemplos Extras

Solução: As relações R_2 e R_3 são simétricas, pois, em cada caso, (b, a) pertence à relação sempre que (a, b) pertencer. Para R_2, a única coisa a verificar é que tanto (2, 1) quanto (1, 2) estão na relação. Para R_3, é necessário verificar que (1, 2) e (2, 1) pertencem à relação e (1, 4) e (4, 1) pertencem à relação. O leitor deve verificar que nenhuma das outras relações é simétrica. Isso é feito encontrando-se um par (a, b) tal que ele esteja na relação, mas (b, a) não.

R_4, R_5 e R_6 são todas anti-simétricas. Para cada uma dessas relações não existe nenhum par de elementos a e b com $a \neq b$, tal que tanto (a, b) como (b, a) pertençam à relação. O leitor deve verificar que nenhuma das outras relações é anti-simétrica. Isso é feito encontrando-se um par (a, b) com $a \neq b$, tal que (a, b) e (b, a) estejam ambos na relação. ◀

EXEMPLO 11 Quais das relações do Exemplo 5 são simétricas e quais são anti-simétricas?

Solução: As relações R_3, R_4 e R_6 são simétricas. R_3 é simétrica, pois se $a = b$ ou $a = -b$, então $b = a$ ou $b = -a$. R_4 é simétrica, pois $a = b$ implica que $b = a$. R_6 é simétrica, pois $a + b \leq 3$ implica que $b + a \leq 3$. O leitor deve verificar que nenhuma das outras relações é simétrica.

As relações R_1, R_2, R_4 e R_5 são anti-simétricas. R_1 é anti-simétrica porque as desigualdades $a \leq b$ e $b \leq a$ implicam que $a = b$. R_2 é anti-simétrica porque é impossível $a > b$ e $b > a$. R_4 é anti-simétrica porque dois elementos são relacionados relativamente a R_4 se e somente se eles forem iguais. R_5 é anti-simétrica porque é impossível $a = b + 1$ e $b = a + 1$. O leitor deve verificar que nenhuma das outras relações é anti-simétrica. ◀

EXEMPLO 12 A relação "divide", no conjunto dos inteiros positivos, é simétrica? É anti-simétrica?

Solução: Essa relação não é simétrica porque $1 \mid 2$, mas $2 \nmid 1$. Ela é anti-simétrica, pois se a e b forem inteiros positivos com $a \mid b$ e $b \mid a$, então $a = b$ (essa verificação é deixada como um exercício para o leitor). ◀

Seja R a relação que consiste em todos os pares (x, y) de estudantes de sua escola, em que x tenha adquirido mais créditos do que y. Suponha que x esteja relacionado a y e y esteja relacionado a z. Isso significa que x adquiriu mais créditos que y e y adquiriu mais créditos que z. Podemos concluir que x adquiriu mais créditos do que z, de modo que x está relacionado a z. O que nós mostramos é que R tem a propriedade transitiva, que é definida como segue.

DEFINIÇÃO 5 Uma relação R em um conjunto A é chamada de *transitiva* se, sempre que $(a, b) \in R$ e $(b, c) \in R$, então $(a, c) \in R$, para todo $a, b, c \in A$.

Lembre-se: Usando quantificadores, vemos que a relação R em um conjunto A é transitiva se tivermos $\forall a \forall b \forall c\, (((a, b) \in R \wedge (b, c) \in R) \rightarrow (a, c) \in R)$.

EXEMPLO 13 Quais das relações do Exemplo 7 são transitivas?

Exemplos Extras

Solução: R_4, R_5 e R_6 são transitivas. Para cada uma dessas relações, podemos mostrar que ela é transitiva verificando que se (a, b) e (b, c) pertencem a essa relação, então (a, c) também pertence. Por exemplo, R_4 é transitiva porque (3, 2) e (2, 1), (4, 2) e (2, 1), (4, 3) e (3, 1), e (4, 3) e (3, 2) são os únicos pares deste tipo, e (3, 1), (4, 1) e (4, 2) pertencem a R_4. O leitor deve verificar que R_5 e R_6 são transitivas.

R_1 não é transitiva porque (3, 4) e (4, 1) pertencem a R_1, mas (3, 1) não pertence. R_2 não é transitiva porque (2, 1) e (1, 2) pertencem a R_2, mas (2, 2) não pertence. R_3 não é transitiva porque (4, 1) e (1, 2) pertencem a R_3, mas (4, 2) não pertence. ◀

EXEMPLO 14 Quais das relações do Exemplo 5 são transitivas?

Solução: As relações R_1, R_2, R_3 e R_4 são transitivas. R_1 é transitiva porque $a \leq b$ e $b \leq c$ implicam que $a \leq c$. R_2 é transitiva porque $a > b$ e $b > c$ implicam que $a > c$. R_3 é transitiva porque $a = \pm b$ e $b = \pm c$ implicam que $a = \pm c$. R_4 é claramente transitiva, como o leitor deve verificar. R_5 não é transitiva porque (2, 1) e (1, 0) pertencem a R_5, mas (2, 0) não pertence. R_6 não é transitiva porque (2, 1) e (1, 2) pertencem a R_6, mas (2, 2) não pertence. ◀

EXEMPLO 15 A relação "divide", no conjunto dos inteiros positivos, é transitiva?

Solução: Suponha que a divida b e que b divida c. Então, existem inteiros positivos k e l tal que $b = ak$ e $c = bl$. Portanto, $c = a(kl)$, de modo que a divide c. Segue que esta relação é transitiva. ◀

Podemos usar técnicas de contagem para determinar o número de relações com propriedades específicas. Encontrar o número de relações com uma propriedade particular fornece informação sobre quão comum essa propriedade é no conjunto de todas as relações em um conjunto com n elementos.

EXEMPLO 16 Quantas relações reflexivas existem em um conjunto com n elementos?

Solução: Uma relação R em um conjunto A é um subconjunto de $A \times A$. Conseqüentemente, uma relação é determinada especificando se cada um dos n^2 pares ordenados em $A \times A$ está em R. Entretanto, se R for reflexiva, cada um dos n pares ordenados (a, a) para $a \in A$, deve estar em R. Cada um dos outros $n(n-1)$ pares ordenados da forma (a, b), com $a \neq b$, pode estar ou não em R. Logo, pela regra do produto para contagem, existem $2^{n(n-1)}$ relações reflexivas [este é o número de maneiras de escolher se cada elemento (a, b), com $a \neq b$, pertence a R]. ◀

O número de relações simétricas e o número de relações anti-simétricas em um conjunto com n elementos podem ser encontrados usando uma argumentação semelhante à do Exemplo 16 (ver o Exercício 45 no final desta seção). Contar as relações transitivas em um conjunto com n elementos é um problema além do escopo deste livro.

Combinando Relações

Como as relações de A em B são subconjuntos de $A \times B$, duas relações de A em B podem ser combinadas de qualquer maneira que dois conjuntos possam ser combinados. Considere os exemplos 17 a 19.

EXEMPLO 17 Seja A $= \{1, 2, 3\}$ e $B = \{1, 2, 3, 4\}$. As relações $R_1 = \{(1, 1), (2, 2), (3, 3)\}$ e $R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4)\}$ podem ser combinadas para se obter

$$R_1 \cup R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (3, 3)\},$$
$$R_1 \cap R_2 = \{(1, 1)\},$$
$$R_1 - R_2 = \{(2, 2), (3, 3)\},$$
$$R_2 - R_1 = \{(1, 2), (1, 3), (1, 4)\}.$$ ◀

EXEMPLO 18 Sejam A e B o conjunto de todos os estudantes e o conjunto de todas as disciplinas em uma escola, respectivamente. Suponha que R_1 consista em todos os pares ordenados (a, b), nos quais a seja um estudante que fez a disciplina b, e R_2 consista em todos os pares ordenados (a, b), nos quais a é um estudante que precisa da disciplina b para se formar. O que são as relações $R_1 \cup R_2$, $R_1 \cap R_2$, $R_1 \oplus R_2$, $R_1 - R_2$ e $R_2 - R_1$?

Solução: A relação $R_1 \cup R_2$ consiste em todos os pares ordenados (a, b), nos quais a é um estudante que ou fez a disciplina b ou precisa da disciplina b para se formar, e $R_1 \cap R_2$ é o conjunto de todos os pares ordenados (a, b), nos quais a é um estudante que fez a disciplina b e que precisa dela para se formar. Além disso, $R_1 \oplus R_2$ consiste em todos os pares ordenados (a, b), nos quais o estudante a fez a disciplina b, mas não precisa dela para se formar, ou precisa da disciplina b para se formar, mas ainda não a fez. $R_1 - R_2$ é o conjunto dos pares ordenados (a, b), nos quais a fez a disciplina b, mas não precisa dela para se formar; isto é, b é uma disciplina optativa que a fez. $R_2 - R_1$ é o conjunto de todos os pares ordenados (a, b), nos quais b é uma disciplina de que a precisa para se formar, mas que ainda não a fez. ◀

EXEMPLO 19 Seja R_1 a relação "menor que" no conjunto dos números reais, e seja R_2 a relação "maior que" no conjunto dos números reais, isto é, $R_1 = \{(x, y) \mid x < y\}$ e $R_2 = \{(x, y) \mid x > y\}$. O que são $R_1 \cup R_2$, $R_1 \cap R_2$, $R_1 - R_2$, $R_2 - R_1$ e $R_1 \oplus R_2$?

Solução: Observamos que $(x, y) \in R_1 \cup R_2$ se e somente se $(x, y) \in R_1$ ou $(x, y) \in R_2$. Logo, $(x, y) \in R_1 \cup R_2$ se e somente se $x < y$ ou $x > y$. Como a condição $x < y$ ou $x > y$ é a mesma que a condição $x \neq y$, segue que $R_1 \cup R_2 = \{(x, y) \mid x \neq y\}$.Em outras palavras, a união da relação "menor que" e da relação "maior que" é a relação "diferente de".

A seguir, observe que é impossível um par (x, y) pertencer a ambas R_1 e R_2, pois é impossível que $x < y$ e $x > y$. Segue que $R_1 \cap R_2 = \emptyset$. Vemos também que $R_1 - R_2 = R_1$, $R_2 - R_1 = R_2$ e $R_1 \oplus R_2 = R_1 \cup R_2 - R_1 \cap R_2 = \{(x, y) \mid x \neq y\}$. ◀

Há uma outra maneira de combinar relações que é análoga à composição de funções.

DEFINIÇÃO 6 Seja R uma relação de um conjunto A em um conjunto B e S uma relação de B em um conjunto C. A *composta* de R e S é a relação que consiste dos pares ordenados (a, c), nos quais $a \in A$ e $c \in C$, e para os quais existe um elemento $b \in B$, tal que $(a, b) \in R$ e $(b, c) \in S$. Indicamos a composta de R e S por $S \circ R$.

Calcular a composta de duas relações exige que encontremos elementos que são segundos elementos de pares ordenados da primeira relação e primeiros elementos de pares ordenados da segunda relação, como ilustram os exemplos 20 e 21.

EXEMPLO 20 O que é a composta das relações R e S, em que R é a relação de $\{1, 2, 3\}$ em $\{1, 2, 3, 4\}$, com $R = \{(1, 1), (1, 4), (2, 3), (3, 1), (3, 4)\}$, e S é a relação de $\{1, 2, 3, 4\}$ em $\{0, 1, 2\}$, com $S = \{(1, 0), (2, 0), (3, 1), (3, 2), (4, 1)\}$?

Solução: $S \circ R$ é construída usando todos os pares ordenados em R e todos os pares ordenados em S, em que o segundo elemento do par ordenado em R coincide com o primeiro elemento do par ordenado em S. Por exemplo, os pares ordenados $(2, 3)$ em R e $(3, 1)$ em S produzem o par ordenado $(2, 1)$ em $S \circ R$. Calculando todos os pares ordenados na composta, encontramos

$$S \circ R = \{(1, 0), (1, 1), (2, 1), (2, 2), (3, 0), (3, 1)\}.$$ ◀

EXEMPLO 21 **Compondo a Relação de Progenitor com Si Mesma** Seja R a relação no conjunto de todas as pessoas, tal que $(a, b) \in R$ se a pessoa a for um progenitor da pessoa b. Então, $(a, c) \in R \circ R$ se e somente se existir uma pessoa b, tal que $(a, b) \in R$ e $(b, c) \in R$, ou seja, se e somente se existir uma pessoa b, tal que a seja progenitor de b e b seja progenitor de c. Em outras palavras, $(a, c) \in R \circ R$ se e somente se a for avô ou avó de c. ◀

As potências de uma relação R podem ser recursivamente definidas a partir da definição de uma composta de duas relações.

DEFINIÇÃO 7 Seja R uma relação em um conjunto A. As potências R^n, $n = 1, 2, 3, \ldots$, são definidas recursivamente por

$$R^1 = R \quad \text{e} \quad R^{n+1} = R^n \circ R.$$

A definição mostra que $R^2 = R \circ R$, $R^3 = R^2 \circ R = (R \circ R) \circ R$, e assim por diante.

EXEMPLO 22 Seja $R = \{(1, 1), (2, 1), (3, 2), (4, 3)\}$. Encontre as potências R^n, $n = 2, 3, 4, ...$

Solução: Como $R^2 = R \circ R$, encontramos que $R^2 = \{(1, 1), (2, 1), (3, 1), (4, 2)\}$. Além disso, como $R^3 = R^2 \circ R$, $R^3 = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$. Cálculos adicionais mostram que R^4 é o mesmo que R^3, de modo que $R^4 = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$. Segue também que $R^n = R^3$ para $n = 5, 6, 7, ...$ O leitor deve verificar isso. ◀

O teorema a seguir mostra que as potências de uma relação transitiva são subconjuntos dessa relação. Ele será usado na Seção 8.4.

TEOREMA 1 A relação R em um conjunto A é transitiva se e somente se $R^n \subseteq R$ para $n = 1, 2, 3, ...$

Demonstração: Demonstramos primeiro a parte "se" do teorema. Suponhamos que $R^n \subseteq R$ para $n = 1, 2, 3, ...$ Em particular, $R^2 \subseteq R$. Para ver que isso implica que R é transitiva, observe que se $(a, b) \in R$ e $(b, c) \in R$, então, por definição de composição, $(a, c) \in R^2$. Como $R^2 \subseteq R$, isso significa que $(a, c) \in R$. Portanto, R é transitiva.

Vamos usar indução matemática para demonstrar a parte "somente se" do teorema. Observe que essa parte do teorema é trivialmente verdadeira para $n = 1$.

Suponha que $R^n \subseteq R$, em que n é um inteiro positivo. Esta é a hipótese de indução. Para completar o passo de indução, devemos mostrar que isso implica que R^{n+1} também é um subconjunto de R. Para mostrar isso, suponha que $(a, b) \in R^{n+1}$. Então, como $R^{n+1} = R^n \circ R$, existe um elemento x com $x \in A$, tal que $(a, x) \in R$ e $(x, b) \in R^n$. A hipótese de indução, ou seja, que $R^n \subseteq R$, implica que $(x, b) \in R$. Além disso, como R é transitiva e $(a, x) \in R$ e $(x, b) \in R$, segue que $(a, b) \in R$. Isso mostra que $R^{n+1} \subseteq R$, completando a demonstração. ◁

Exercícios

1. Liste os pares ordenados na relação R de $A = \{0, 1, 2, 3, 4\}$ em $B = \{0, 1, 2, 3\}$, em que $(a, b) \in R$ se e somente se
- **a)** $a = b$.
- **b)** $a + b = 4$.
- **c)** $a > b$.
- **d)** $a \mid b$.
- **e)** $\text{mdc}(a, b) = 1$.
- **f)** $\text{mmc}(a, b) = 2$.

2. **a)** Liste todos os pares ordenados na relação $R = \{(a, b) \mid a \text{ divide } b\}$ no conjunto $\{1, 2, 3, 4, 5, 6\}$.
- **b)** Mostre essa relação graficamente, como foi feito no Exemplo 4.
- **c)** Mostre essa relação na forma tabular, como foi feito no Exemplo 4.

3. Para cada uma destas relações no conjunto $\{1, 2, 3, 4\}$, decida se ela é reflexiva, se é simétrica, se é anti-simétrica e se é transitiva.
- **a)** $\{(2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)\}$
- **b)** $\{(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (4, 4)\}$
- **c)** $\{(2, 4), (4, 2)\}$
- **d)** $\{(1, 2), (2, 3), (3, 4)\}$
- **e)** $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$
- **f)** $\{(1, 3), (1, 4), (2, 3), (2, 4), (3, 1), (3, 4)\}$

4. Determine se a relação R no conjunto de todas as pessoas é reflexiva, simétrica, anti-simétrica e/ou transitiva, em que $(a, b) \in R$ se e somente se
- **a)** a é mais alto que b.
- **b)** a e b nasceram no mesmo dia.
- **c)** a tem o mesmo prenome que b.
- **d)** a e b tem um avô ou avó em comum.

5. Determine se a relação R no conjunto de todas as páginas da Web é reflexiva, simétrica, anti-simétrica e/ou transitiva, em que $(a, b) \in R$ se e somente se
- **a)** todo mundo que visitou a página a da Web também visitou a página b da Web.
- **b)** não há links comuns encontrados tanto na página a da Web quanto na página b da Web.
- **c)** existe, pelo menos, um link comum na página a da Web e na página b da Web.
- **d)** existe uma página da Web que inclui links para ambas as páginas a e b da Web.

6. Determine se a relação R no conjunto de todos os números reais é reflexiva, simétrica, anti-simétrica e/ou transitiva, em que $(x, y) \in R$ se e somente se

a) $x + y = 0$. **b)** $x = \pm y$.
c) $x - y$ é um número racional. **d)** $x = 2y$.
e) $xy \geq 0$. **f)** $xy = 0$.
g) $x = 1$. **h)** $x = 1$ ou $y = 1$.

7. Determine se a relação R no conjunto de todos os inteiros é reflexiva, simétrica, anti-simétrica e/ou transitiva, em que $(x, y) \in R$ se e somente se
a) $x \neq y$. **b)** $xy \geq 1$.
c) $x = y + 1$ ou $x = y - 1$.
d) $x \equiv y \pmod 7$. **e)** x é um múltiplo de y.
f) x e y são ambos negativos ou ambos não negativos.
g) $x = y^2$. **h)** $x \geq y^2$.

8. Dê um exemplo de uma relação em um conjunto que seja
a) simétrica e anti-simétrica.
b) nem simétrica nem anti-simétrica.

Uma relação R em um conjunto A é **irreflexiva** se para todo $a \in A$, $(a, a) \notin R$. Isto é, R é irreflexiva se nenhum elemento de A for relacionado a si próprio.

9. Quais relações no Exercício 3 são irreflexivas?
10. Quais relações no Exercício 4 são irreflexivas?
11. Quais relações no Exercício 5 são irreflexivas?
12. Quais relações no Exercício 6 são irreflexivas?
13. Uma relação em um conjunto pode não ser nem reflexiva nem irreflexiva?
14. Use quantificadores para expressar o que significa uma relação ser irreflexiva.
15. Dê um exemplo de uma relação irreflexiva no conjunto de todas as pessoas.

Uma relação R é chamada de **assimétrica** se $(a, b) \in R$ implica que $(b, a) \notin R$. Os exercícios 16 a 22 exploram a noção de uma relação assimétrica. O Exercício 20 se concentra na diferença entre assimetria e anti-simetria.

16. Quais relações no Exercício 3 são assimétricas?
17. Quais relações no Exercício 4 são assimétricas?
18. Quais relações no Exercício 5 são assimétricas?
19. Quais relações no Exercício 6 são assimétricas?
20. Uma relação assimétrica precisa ser também anti-simétrica? Uma relação anti-simétrica precisa ser também assimétrica? Dê razões para sua resposta.
21. Use quantificadores para expressar o que significa uma relação ser assimétrica.
22. Dê um exemplo de uma relação assimétrica no conjunto de todas as pessoas.
23. Quantas relações diferentes existem de um conjunto com m elementos em um conjunto com n elementos?

☞Seja R uma relação de um conjunto A em um conjunto B. A **relação inversa** de B para A, indicada por R^{-1}, é o conjunto dos pares ordenados $\{(b, a) \mid (a, b) \in R\}$. A **relação complementar** $\overline{R}$ é o conjunto dos pares ordenados $\{(a, b) \mid (a, b) \notin R\}$.

24. Seja R a relação $R = \{(a, b) \mid a < b\}$ no conjunto dos inteiros. Encontre
a) R^{-1}. **b)** $\overline{R}$.

25. Seja R a relação $R = \{(a, b) \mid a \text{ divide } b\}$ no conjunto dos inteiros positivos. Encontre
a) R^{-1}. **b)** $\overline{R}$.

26. Seja R a relação no conjunto de todos os estados dos Estados Unidos que consiste em pares (a, b), em que o estado a faz fronteira com o estado b. Encontre
a) R^{-1}. **b)** $\overline{R}$.

27. Suponha que a função f de A para B seja uma correspondência injetora. Seja R a relação que coincide com o gráfico de f. Isto é, $R = \{(a, f(a)) \mid a \in A\}$. O que é a relação inversa R^{-1}?

28. Sejam $R_1 = \{(1, 2), (2, 3), (3, 4)\}$ e $R_2 = \{(1, 1), (1, 2), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (3, 4)\}$ relações de $\{1, 2, 3\}$ em $\{1, 2, 3, 4\}$. Encontre
a) $R_1 \cup R_2$. **b)** $R_1 \cap R_2$.
c) $R_1 - R_2$. **d)** $R_2 - R_1$.

29. Sejam A o conjunto dos estudantes de sua escola e B o conjunto dos livros na biblioteca da escola. Sejam R_1 e R_2 as relações que consistem em todos os pares ordenados (a, b), nos quais é exigido que o estudante a leia o livro b em uma disciplina, e nos quais o estudante a leu o livro b, respectivamente. Descreva os pares ordenados em cada uma destas aplicações.
a) $R_1 \cup R_2$ **b)** $R_1 \cap R_2$
c) $R_1 \oplus R_2$ **d)** $R_1 - R_2$
e) $R_2 - R_1$

30. Seja R a relação $\{(1, 2), (1, 3), (2, 3), (2, 4), (3, 1)\}$ e seja S a relação $\{(2, 1), (3, 1), (3, 2), (4, 2)\}$. Determine $S \circ R$.

31. Seja R a relação no conjunto das pessoas que consiste nos pares (a, b), nos quais a é progenitor de b. Seja S a relação no conjunto das pessoas que consiste nos pares (a, b), nos quais a e b são irmãos. O que são $S \circ R$ e $R \circ S$?

Os exercícios 32 a 35 tratam destas relações no conjunto dos números reais:

$R_1 = \{(a, b) \in \mathbf{R}^2 \mid a > b\}$, a relação "maior que",
$R_2 = \{(a, b) \in \mathbf{R}^2 \mid a \geq b\}$, a relação "maior que ou igual a",

$R_3 = \{(a, b) \in \mathbf{R}^2 \mid a < b\}$, a relação "menor que",
$R_4 = \{(a, b) \in \mathbf{R}^2 \mid a \leq b\}$, a relação "menor que ou igual a",

$R_5 = \{(a, b) \in \mathbf{R}^2 \mid a = b\}$, a relação "igual a",
$R_6 = \{(a, b) \in \mathbf{R}^2 \mid a \neq b\}$, a relação "diferente de".

32. Determine
a) $R_1 \cup R_3$. **b)** $R_1 \cup R_5$.
c) $R_2 \cap R_4$. **d)** $R_3 \cap R_5$.
e) $R_1 - R_2$. **f)** $R_2 - R_1$.
g) $R_1 \oplus R_3$. **h)** $R_2 \oplus R_4$.

33. Determine
a) $R_2 \cup R_4$. **b)** $R_3 \cup R_6$.
c) $R_3 \cap R_6$. **d)** $R_4 \cap R_6$.
e) $R_3 - R_6$. **f)** $R_6 - R_3$.
g) $R_2 \oplus R_6$. **h)** $R_3 \oplus R_5$.

34. Determine
a) $R_1 \circ R_1$. **b)** $R_1 \circ R_2$.
c) $R_1 \circ R_3$. **d)** $R_1 \circ R_4$.
e) $R_1 \circ R_5$. **f)** $R_1 \circ R_6$.
g) $R_2 \circ R_3$. **h)** $R_3 \circ R_3$.

35. Determine
a) $R_2 \circ R_1$. **b)** $R_2 \circ R_2$.
c) $R_3 \circ R_5$. **d)** $R_4 \circ R_1$.
e) $R_5 \circ R_3$. **f)** $R_3 \circ R_6$.
g) $R_4 \circ R_6$. **h)** $R_6 \circ R_6$.

36. Seja R a relação de progenitor no conjunto de todas as pessoas (ver o Exemplo 21). Quando um par ordenado está na relação R^3?

37. Seja R a relação no conjunto das pessoas com doutorado, tal que $(a, b) \in R$ se e somente se a foi o orientador de tese de b. Quando um par ordenado (a, b) está em R^2? Quando um par ordenado (a, b) está em R^n, quando n é um inteiro positivo? (Observe que toda pessoa com doutorado tem um orientador de tese.)

38. Sejam R_1 e R_2 as relações "divide" e "é múltiplo de" no conjunto de todos os inteiros positivos, respectivamente. Ou seja, $R_1 = \{(a, b) \mid a \text{ divide } b\}$ e $R_2 = \{(a, b) \mid a \text{ é múltiplo de } b\}$. Determine
a) $R_1 \cup R_2$. **b)** $R_1 \cap R_2$.
c) $R_1 - R_2$. **d)** $R_2 - R_1$.
e) $R_1 \oplus R_2$.

39. Sejam R_1 e R_2 as relações "congruente módulo 3" e "congruente módulo 4", respectivamente, no conjunto dos inteiros. Isto é, $R_1 = \{(a, b) \mid a \equiv b \pmod 3\}$ e $R_2 = \{(a, b) \mid a \equiv b \pmod 4\}$. Determine
a) $R_1 \cup R_2$. **b)** $R_1 \cap R_2$.
c) $R_1 - R_2$. **d)** $R_2 - R_1$.
e) $R_1 \oplus R_2$.

40. Liste as 16 relações diferentes no conjunto $\{0, 1\}$.

41. Quantas das 16 relações diferentes em $\{0, 1\}$ contêm o par $(0, 1)$?

42. Quais das 16 relações em $\{0, 1\}$, que você listou no Exercício 40, são
a) reflexivas?
b) irreflexivas?
c) simétricas?
d) anti-simétricas?
e) assimétricas?
f) transitivas?

43. **a)** Quantas relações existem no conjunto $\{a, b, c, d\}$?
b) Quantas relações existem no conjunto $\{a, b, c, d\}$ que contêm o par (a, a)?

44. Seja S um conjunto com n elementos e sejam a e b elementos distintos de S. Quantas relações existem em S, tal que
a) $(a, b) \in S$? **b)** $(a, b) \notin S$?
c) não existe nenhum par ordenado na relação que tenha a como primeiro elemento?
d) existe pelo menos um par ordenado na relação que tenha a como primeiro elemento?
e) não existe nenhum par ordenado na relação que tem a como primeiro elemento e não existe nenhum par ordenado na relação que tenha b como segundo elemento?
f) existe pelo menos um par ordenado na relação que ou tenha a como primeiro elemento ou b como segundo elemento?

***45.** Quantas relações existem em um conjunto com n elementos que sejam
a) simétricas?
b) anti-simétricas?
c) assimétricas?
d) irreflexivas?
e) reflexivas e simétricas?
f) nem reflexivas nem irreflexivas?

***46.** Quantas relações transitivas existem em um conjunto com n elementos se
a) $n = 1$? **b)** $n = 2$? **c)** $n = 3$?

47. Encontre o erro na "demonstração" do seguinte "teorema".
"Teorema": Seja R uma relação em um conjunto A, que é simétrica e transitiva. Então R é reflexiva.
"Demonstração": Seja $a \in A$. Considere um elemento $b \in A$, tal que $(a, b) \in R$. Como R é simétrica, temos também que $(b, a) \in R$. Agora, usando a propriedade transitiva, podemos concluir que $(a, a) \in R$, pois $(a, b) \in R$ e $(b, a) \in R$.

48. Suponha que R e S sejam relações reflexivas em um conjunto A. Demonstre ou dê contra-exemplo para cada uma destas afirmações.
a) $R \cup S$ é reflexiva.
b) $R \cap S$ é reflexiva.
c) $R \oplus S$ é irreflexiva.
d) $R - S$ é irreflexiva.
e) $S \circ S$ é reflexiva.

49. Mostre que a relação R em um conjunto A é simétrica se e somente se $R = R^{-1}$, em que R^{-1} é a relação inversa.

50. Mostre que a relação R em um conjunto A é anti-simétrica se e somente se $R \cap R^{-1}$ for um subconjunto da relação diagonal $\Delta = \{(a, a) \mid a \in A\}$.

51. Mostre que a relação R em um conjunto A é reflexiva se e somente se a relação inversa R^{-1} for reflexiva.

52. Mostre que a relação R em um conjunto A é reflexiva se e somente se a relação complementar $\overline{R}$ for irreflexiva.

53. Seja R uma relação que é reflexiva e transitiva. Demonstre que $R^n = R$ para todo inteiro positivo n.

54. Seja R a relação no conjunto $\{1, 2, 3, 4, 5\}$ com os pares ordenados $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 3)$, $(2, 4)$, $(3, 1)$, $(3, 4)$, $(3, 5)$, $(4, 2)$, $(4, 5)$, $(5, 1)$, $(5, 2)$ e $(5, 4)$. Determine
a) R^2. **b)** R^3. **c)** R^4. **d)** R^5.

55. Seja R uma relação reflexiva em um conjunto A. Mostre que R^n é reflexiva para todo inteiro positivo n.

***56.** Seja R uma relação simétrica. Mostre que R^n é simétrica para todo inteiro positivo n.

57. Suponha que a relação R seja irreflexiva. R^2 é necessariamente irreflexiva? Dê uma razão para sua resposta.

8.2 Relações *n*-árias e Suas Aplicações

Introdução

Relações entre elementos de mais de dois conjuntos aparecem com freqüência. Por exemplo, existe uma relação que envolve o nome de um estudante, o curso do estudante e sua média geral. Analogamente, existe uma relação que envolve a companhia aérea, o número do vôo, o ponto inicial, o destino, a hora de partida e a hora de chegada do vôo. Um exemplo dessa relação na matemática envolve três inteiros, na qual o primeiro inteiro é maior do que o segundo inteiro, que é maior do que o terceiro. Outro exemplo é a relação de estar "entre", que envolve pontos em uma reta, de modo que três pontos estão relacionados quando o segundo ponto estiver entre o primeiro e o terceiro.

Estudaremos relações entre elementos de mais de dois conjuntos nesta sessão. Essas relações são chamadas de **relações *n*-árias**. Elas são usadas para representar banco de dados no computador. Essas representações nos ajudam a responder questões sobre a informação armazenada no banco de dados, tais como: quais vôos aterrizam no aeroporto O'Hare entre três e quatro horas da manhã? Quais estudantes na sua escola são do último ano e estão se formando em matemática ou ciência da computação e têm média geral maior do que 3,0? Quais empregados de uma companhia trabalharam para ela menos do que cinco anos e ganham mais de $ 50 mil?

Relações *n*-árias

Começamos com a definição básica sobre a qual se apóia a teoria dos bancos de dados relacionais.

DEFINIÇÃO 1 Sejam $A_1, A_2, \ldots A_n$ conjuntos. Uma *relação n-ária* nesses conjuntos é um subconjunto de $A_1 \times A_2 \times \ldots \times A_n$. Os conjuntos $A_1, A_2, \ldots, A_n$ são chamados de *domínios* da relação e n é chamado de seu *grau*.

EXEMPLO 1 Seja R a relação em $\mathbf{N} \times \mathbf{N} \times \mathbf{N}$ que consiste em trios (a, b, c), em que a, b e c são inteiros, com $a < b < c$. Então $(1, 2, 3) \in R$, mas $(2, 4, 3) \notin R$. O grau dessa relação é três. Seus domínios são todos iguais ao conjunto dos números naturais. ◀

EXEMPLO 2 Seja R a relação em $\mathbf{Z} \times \mathbf{Z} \times \mathbf{Z}$ que consiste em todos os trios de inteiros (a, b, c), nos quais a, b e c formam uma progressão aritmética. Ou seja, $(a, b, c) \in R$ se e somente se existir um inteiro k, tal que $b = a + k$ e $c = a + 2k$, ou equivalentemente, tal que $b - a = k$ e $c - b = k$. Observe que $(1, 3, 5) \in R$, pois $3 = 1 + 2$ e $5 = 1 + 2 \cdot 2$, mas $(2, 5, 9) \notin R$, pois $5 - 2 = 3$ enquanto $9 - 5 = 4$. Essa relação tem grau três e seus domínios são todos iguais ao conjunto dos inteiros. ◀

EXEMPLO 3 Seja R a relação em $\mathbf{Z} \times \mathbf{Z} \times \mathbf{Z}^+$, que consiste em trios (a, b, m), em que a, b e m são inteiros, com $m \geq 1$ e $a \equiv b \pmod{m}$. Então, $(8, 2, 3)$, $(-1, 9, 5)$ e $(14, 0, 7)$, todos pertencem a R, mas $(7, 2, 3)$, $(-2, -8, 5)$ e $(11, 0, 6)$ não pertencem a R, pois $8 \equiv 2 \pmod{3}$, $-1 \equiv 9 \pmod{5}$ e $14 \equiv 0 \pmod{7}$, mas $7 \not\equiv 2 \pmod{3}$, $-2 \not\equiv -8 \pmod{5}$ e $11 \not\equiv 0 \pmod{6}$. Essa relação tem grau 3, e seus primeiros dois domínios são o conjunto de todos os inteiros e seu terceiro domínio é o conjunto dos inteiros positivos. ◀

EXEMPLO 4 Seja R a relação que consiste em 5-uplas (A, N, S, D, T) que representam vôos, em que A é a companhia aérea, N é o número do vôo, S é o ponto de partida, D é o destino e T é o horário de partida. Por exemplo, se o Expresso Aéreo Nadir tiver o vôo 963 de Newark para Bangor às 15 horas, então (Nadir, 963, Newark, Bangor, 15 horas) pertence a R. O grau dessa relação é 5, e seus

domínios são o conjunto de todas as companhias aéreas, o conjunto de todos os números dos vôos, o conjunto das cidades, o conjunto das cidades (novamente) e o conjunto dos horários. ◀

Banco de Dados e Relações

O tempo necessário para manipular informação em um banco de dados depende de como essa informação está armazenada. As operações de adicionar e deletar registros, atualizar registros, buscar registros e combinar registros de bancos de dados sobrepostos são feitas milhões de vezes todos os dias em um banco de dados grande. Por causa da importância dessas operações, foram desenvolvidos vários métodos para representar bancos de dados. Discutiremos um desses métodos, chamado de **modelo relacional de dados**, que se baseia no conceito de uma relação.

Um banco de dados consiste em **registros**, os quais são *n*-uplas formadas por **campos**. Os campos são as componentes das *n*-uplas. Por exemplo, um banco de dados com notas de estudantes pode ser formado de campos com o nome, o número do estudante, o curso e a média geral do estudante. O modelo relacional de dados representa um banco de dados de registros como uma relação *n*-ária. Assim, os registros dos estudantes são representados como 4-uplas da forma (*NOME DO ESTUDANTE, NÚMERO DE IDENTIFICAÇÃO, CURSO, MÉDIA GERAL*). Um exemplo de banco de dados com seis desses registros é

(Ackermann, 231455, Ciência da Computação, 3,88)
(Adams, 888323, Física, 3,45)
(Chou, 102147, Ciência da Computação, 3,49)
(Goodfriend, 453876, Matemática, 3,45)
(Rao, 678543, Matemática, 3,90)
(Stevens, 786576, Psicologia, 2,99).

As relações usadas para representar bancos de dados são também chamadas de **tabelas**, porque essas relações são freqüentemente mostradas como tabelas. Cada coluna da tabela corresponde a um *atributo* do banco de dados. Por exemplo, o mesmo banco de dados dos estudantes é mostrado na Tabela 1. Os atributos desse banco de dados são Nome do Estudante, Número de Identificação, Curso e Média Geral.

O domínio de uma relação *n*-ária é chamado de **chave primária** quando o valor desse domínio na *n*-upla determina a *n*-upla. Isto é, o domínio é uma chave primária quando quaisquer duas *n*-uplas distintas na relação têm valores distintos nesse domínio.

Registros são adicionados ou deletados com freqüência dos bancos de dados. Por essa razão, a propriedade de que um domínio seja uma chave primária depende do tempo. Conseqüentemente, deve ser escolhida uma chave primária que permaneça assim sempre que o banco de dados for mudado. A coleção corrente de *n*-uplas em uma relação é chamada de **extensão** da relação. A parte mais permanente do banco de dados, incluindo o nome e os atributos do banco de dados, é chamada sua **intensão**. Ao escolher uma chave primária, o objetivo deveria ser escolher uma chave que possa servir como chave primária para todas as extensões possíveis do banco de dados. Para fazer isso, é necessário examinar a intensão do banco de dados para entender o conjunto das possíveis *n*-uplas que podem ocorrer em uma extensão.

TABELA 1 Estudantes.

Nome_do_estudante	*Numero_de_identificacao*	*Curso*	*Media_geral*
Ackermann	231455	Ciência da Computação	3,88
Adams	888323	Física	3,45
Chou	102147	Ciência da Computação	3,49
Goodfriend	453876	Matemática	3,45
Rao	678543	Matemática	3,90
Stevens	786576	Psicologia	2,99

EXEMPLO 5 Quais domínios são chaves primárias para a relação n-ária mostrada na Tabela 1, supondo que nenhuma n-upla será adicionada no futuro?

Solução: Como existe apenas uma 4-upla nessa tabela para cada nome de estudante, o domínio do nome dos estudantes é uma chave primária. Analogamente, os números de identificação nessa tabela são únicos, de modo que o domínio dos números de identificação também é uma chave primária. Entretanto, o domínio dos cursos não é uma chave primária, porque mais do que uma 4-upla contém o mesmo curso. O domínio de média geral também não é uma chave primária, pois existem duas 4-uplas com a mesma média geral. ◀

Combinações de domínios também podem identificar de forma única n-uplas em uma relação n-ária. Quando os valores de um conjunto de domínios determinam uma n-upla em uma relação, o produto cartesiano desses domínios é chamado de **chave composta**.

EXEMPLO 6 O produto cartesiano do domínio do curso e do domínio da média geral é uma chave composta para a relação n-ária da Tabela 1, supondo que nenhuma n-upla jamais será adicionada?

Solução: Como nenhum par de 4-uplas tem tanto o mesmo curso quanto a mesma média geral, este produto cartesiano é uma chave composta. ◀

Como chaves primárias e compostas são usadas para identificar registros de modo único em um banco de dados, é importante que as chaves permaneçam válidas quando novos registros forem adicionados ao banco de dados. Assim, devem ser feitas verificações para garantir que todo novo registro tenha valores que sejam diferentes no campo, ou campos, apropriado, de todos os outros registros da tabela. Por exemplo, faz sentido usar o número de identificação do estudante como uma chave para o registro dos estudantes se nunca dois estudantes tiverem o mesmo número de identificação. Uma universidade não deveria usar o campo do nome como chave, porque dois estudantes podem ter o mesmo nome (como John Smith).

Operações em Relações n-árias

Existe uma variedade de operações em relações n-árias que podem ser usadas para formar novas relações n-árias. Aplicadas juntas, essas operações podem responder questões sobre banco de dados que pedem todas as n-uplas que satisfazem determinadas condições.

A operação mais básica em uma relação n-ária é determinar todas as n-uplas na relação n-ária que satisfaz determinadas condições. Por exemplo, podemos querer encontrar todos os registros de todos os estudantes de ciência da computação em um banco de dados de registros de estudantes. Podemos querer encontrar os registros de todos os estudantes com uma média geral acima de 3,5 no mesmo banco de dados. Podemos querer encontrar os registros de todos os estudantes de ciência da computação que tenham uma média geral acima de 3,5 neste banco de dados. Para realizar essas tarefas, usamos o operador de seleção.

DEFINIÇÃO 2 Seja R uma relação n-ária e C uma condição que os elementos em R podem satisfazer. Então, o *operador de seleção* s_C leva a relação n-ária R à relação n-ária de todas as n-uplas de R que satisfazem à condição C.

EXEMPLO 7 Para encontrar os registros dos estudantes de ciência da computação na relação n-ária R mostrada na Tabela 1, usamos o operador s_{C_1}, em que C_1 é a condição Curso = "Ciência da Computação".

O resultado consiste nas duas 4-uplas (Ackermann, 231455, Ciência da Computação, 3,88) e (Chou, 102147, Ciência da Computação, 3,49). Analogamente, para encontrar os registros dos estudantes que tenham uma média geral maior do que 3,5 neste banco de dados, usamos o operador s_{C_2}, em que C_2 é a condição Média > 3,5. O resultado consiste nas duas 4-uplas (Ackermann, 231455, Ciência da Computação, 3,88) e (Rao, 678543, Matemática, 3,90). Finalmente, para encontrar os registros dos estudantes de Ciência da Computação que tenham média geral acima de 3,5, usamos o operador s_{C_3}, em que C_3 é a condição (Curso = "Ciência da Computação" $\wedge$ Média > 3,5). O resultado consiste na única 4-upla (Ackermann, 231455, Ciência da Computação, 3,88). ◀

Projeções são usadas para formar novas relações *n*-árias removendo os mesmos campos em todos os registros da relação.

DEFINIÇÃO 3 A *projeção* $P_{i_1 i_2, \ldots, i_m}$, em que $i_1 < i_2 < \cdots i_m$, leva a *n*-upla $(a_1, a_2, \ldots, a_n)$ à *m*-upla $(a_{i_1}, a_{i_2}, \ldots, a_{i_m})$, em que $m \leq n$.

Em outras palavras, a projeção $P_{i_1 i_2, \ldots, i_m}$ remove $n - m$ das componentes de uma *n*-upla, deixando as i_1-ésima, i_2 -ésima, ..., e i_m -ésima componentes.

EXEMPLO 8 O que resulta quando a projeção $P_{1,3}$ é aplicada às 4-uplas (2, 3, 0, 4), (Jane Doe, 234111001, Geografia, 3,14) e (a_1, a_2, a_3, a_4)?

Solução: A projeção $P_{1,3}$ manda as 4-uplas em (2, 0), (Jane Doe, Geografia) e (a_1, a_3), respectivamente. ◀

O Exemplo 9 ilustra como novas relações são produzidas usando projeções.

EXEMPLO 9 O que resulta quando a projeção $P_{1,4}$ é aplicada à relação na Tabela 1?

Solução: Quando é usada a projeção $P_{1,4}$, a segunda e a terceira colunas da tabela são removidas e pares que representam os nomes dos estudantes e a média geral dos estudantes são obtidos. A Tabela 2 mostra o resultado desta projeção. ◀

Podem resultar menos linhas quando uma projeção é aplicada à tabela de uma relação. Isso ocorre quando algumas das *n*-uplas na relação têm valores idênticos em cada uma das *m* componentes da projeção e apenas são distintas nas componentes deletadas pela projeção. Como ilustração, considere o seguinte exemplo.

EXEMPLO 10 Qual é a tabela obtida quando a projeção $P_{1,2}$ é aplicada à relação na Tabela 3?

Solução: A Tabela 4 mostra a relação obtida quando $P_{1,2}$ é aplicada à Tabela 3. Observe que existem menos linhas depois dessa projeção ser aplicada. ◀

A operação **conjunção** é usada para combinar duas tabelas em uma quando essas tabelas compartilham alguns campos idênticos. Por exemplo, uma tabela que contém campos para a companhia aérea, o número do vôo e o portão de embarque, e uma outra tabela que contém campos para o número do vôo, o portão de embarque e o horário de partida podem ser combinadas em uma tabela com campos para a companhia aérea, o número do vôo, o portão de embarque e o horário de partida.

TABELA 2 Medias_gerais.

Nome_do_estudante	*Media*
Ackermann	3,88
Adams	3,45
Chou	3,49
Goodfriend	3,45
Rao	3,90
Stevens	2,99

TABELA 3 Matriculas.

Estudante	*Curso*	*Disciplina*
Glauser	Biologia	BI 290
Glauser	Biologia	MS 475
Glauser	Biologia	PY 410
Marcus	Matemática	MS 511
Marcus	Matemática	MS 603
Marcus	Matemática	CS 322
Miller	Ciência da Computação	MS 575
Miller	Ciência da Computação	CS 455

TABELA 4 Cursos.

Estudante	*Curso*
Glauser	Biologia
Marcus	Matemática
Miller	Ciência da Computação

DEFINIÇÃO 4 Sejam R uma relação de grau m e S uma relação de grau n. A *conjunção* $J_p(R, S)$, em que $p \leq m$ e $p \leq n$, é uma relação de grau $m + n - p$ que consiste em todas as $(m + n - p)$-uplas $(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p})$, em que a m-upla $(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p)$ pertence a R e a n-upla $(c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p})$ pertence a S.

Em outras palavras, o operador junção (ou conjunção) J_p produz uma nova relação a partir de duas relações, combinando todas as m-uplas da primeira relação com todas as n-uplas da segunda relação, em que as últimas p componentes das m-uplas coincidem com as primeiras p componentes das n-uplas.

EXEMPLO 11 Que relação resulta quando o operador junção J_2 for usado para combinar as relações mostradas nas tabelas 5 e 6?

Solução: A conjunção J_2 produz a relação mostrada na Tabela 7. ◀

Existem outros operadores além das projeções e das junções que produzem novas relações a partir de relações existentes. Uma descrição dessas operações pode ser encontrada em livros sobre a teoria de banco de dados.

TABELA 5 Atribuicao_de_carga_didatica.

Professor	*Departamento*	*Numero_da_disciplina*
Cruz	Zoologia	335
Cruz	Zoologia	412
Farber	Psicologia	501
Farber	Psicologia	617
Grammer	Física	544
Grammer	Física	551
Rosen	Ciência da Computação	518
Rosen	Matemática	575

TABELA 6 Horario_de_Aulas.

Departamento	*Numero_da_disciplina*	*Sala*	*Horario*
Ciência da Computação	518	N521	2:00 P.M.
Matemática	575	N502	3:00 P.M.
Matemática	611	N521	4:00 P.M.
Física	544	B505	4:00 P.M.
Psicologia	501	A100	3:00 P.M.
Psicologia	617	A110	11:00 A.M.
Zoologia	335	A100	9:00 A.M.
Zoologia	412	A100	8:00 A.M

TABELA 7 Horario_dos_professores.

Professor	*Departamento*	*Numero_da_disciplina*	*Sala*	*Horario*
Cruz	Zoologia	335	A100	9:00 A.M.
Cruz	Zoologia	412	A100	8:00 A.M.
Farber	Psicologia	501	A100	3:00 P.M.
Farber	Psicologia	617	A110	11:00 A.M.
Grammer	Física	544	B505	4:00 P.M.
Rosen	Ciência da Computação	518	N521	2:00 P.M.
Rosen	Matemática	575	N502	3:00 P.M.

SQL

A linguagem de consulta SQL para banco de dados (abreviação de *structured query language*, em inglês) pode ser usada para efetuar as operações que descrevemos nesta seção. O Exemplo 12 ilustra como os comandos SQL estão relacionados com as operações sobre relações *n*-árias.

EXEMPLO 12 Vamos ilustrar como a linguagem SQL é utilizada para expressar uma consulta sobre vôos usando a Tabela 8. O comando SQL

```
SELECT  Horario_de_partida
FROM Voos
WHERE Destino=' Detroit'
```

é usado para encontrar a projeção P_5 (sobre o atributo Horario_de_partida) da seleção de 5-uplas no banco de dados Vôos que satisfazem a condição: Destino ='Detroit'. A saída seria uma lista com os horários dos vôos que têm Detroit como destino, ou seja, 08:10, 08:47 e 09:44. A linguagem SQL usa a cláusula FROM para identificar a relação *n*-ária à qual a consulta é aplicada, a cláusula WHERE para especificar a condição da operação de seleção e a cláusula SELECT para especificar a operação de projeção que deve ser aplicada. (*Cuidado*: SQL usa SELECT para representar uma projeção, em vez de uma operação de seleção. Este é um exemplo infeliz de terminologia conflitante.) ◄

O Exemplo 13 mostra como podem ser feitas consultas SQL que envolvem mais de uma tabela.

TABELA 8 Voos.

Companhia_aerea	*Numero_do_voo*	*Portao*	*Destino*	*Horario_de_partida*
Nadir	122	34	Detroit	08:10
Acme	221	22	Denver	08:17
Acme	122	33	Anchorage	08:22
Acme	323	34	Honolulu	08:30
Nadir	199	13	Detroit	08:47
Acme	222	22	Denver	09:10
Nadir	322	34	Detroit	09:44

EXEMPLO 13 O comando SQL

```
SELECT Professor, Horario
FROM Atribuicao_da_carga_didatica, horario_de_aula
WHERE Departamento =' Matematica'
```

é usado para encontrar a projeção $P_{1,5}$ das 5-uplas no banco de dados (mostrado na Tabela 7), que é a conjunção J_2 dos bancos de dados Atribuicoes_de_carga_didatica e Horario_de_aula nas tabelas 5 e 6, respectivamente, que satisfazem a condição: Departamento = Matematica. A saída consistiria no par único (Rosen, 3:00 PM). A cláusula SQL FROM é usada aqui para encontrar a conjunção de dois bancos de dados diferentes. ◀

Nós apenas mencionamos os conceitos básicos de banco de dados relacionais nesta seção. Mais informações podem ser encontradas em [AhU195].

Exercícios

1. Liste os trios na relação $\{(a, b, c) \mid a, b \text{ e } c \text{ são inteiros, com } 0 < a < b < c < 5\}$.
2. Quais 4-uplas estão na relação $\{(a, b, c, d) \mid a, b, c \text{ e } d \text{ são inteiros positivos, com } abcd = 6\}$?
3. Liste as 5-uplas na relação da Tabela 8.
4. Supondo que nenhuma n-upla nova seja adicionada, encontre todas as chaves primárias para as relações mostradas na
   a) Tabela 3. **b)** Tabela 5.
   c) Tabela 6. **d)** Tabela 8.
5. Supondo que nenhuma n-upla nova seja adicionada, encontre uma chave composta com dois campos que contenham o campo *Companhia_aerea* para o banco de dados da Tabela 8.
6. Supondo que nenhuma n-upla nova seja adicionada, encontre uma chave composta com dois campos que contenham o campo *Professor* para o banco de dados da Tabela 7.
7. As 3-uplas na relação 3-ária representam os seguintes atributos do banco de dados de um estudante: número de identificação do estudante, nome, número do telefone.
   a) É provável que o número de identificação do estudante seja uma chave primária?
   b) É provável que o nome seja uma chave primária?
   c) É provável que o número do telefone seja uma chave primária?
8. As 4-uplas em uma relação 4-ária representam estes atributos de livros publicados: título, ISBN, data da publicação, número de páginas.
   a) O que provavelmente é uma chave primária para essa relação?
   b) Sob quais condições (título, data de publicação) seria uma chave composta?
   c) Sob quais condições (título, número de páginas) seria uma chave composta?
9. As 5-uplas em uma relação 5-ária representam estes atributos de todas as pessoas nos Estados Unidos: nome, número do Social Security, endereço, cidade, estado.
   a) Determine uma chave primária para essa relação.
   b) Sob quais condições (nome, endereço) seria uma chave composta?
   c) Sob quais condições (nome, endereço, cidade) seria uma chave composta?
10. O que você obtém quando aplica a operação seleção s_C, em que C é a condição Sala = A100, ao banco de dados da Tabela 7?
11. O que você obtém quando aplica a operação seleção s_C, em que C é a condição Destino = Detroit, ao banco de dados da Tabela 8?
12. O que você obtém quando aplica a operação seleção s_C, em que C é a condição (Projeto = 2) $\wedge$ (Quantidade $\geq$ 50), ao banco de dados da Tabela 10 (encontrada mais adiante, neste conjunto de exercícios)?
13. O que você obtém quando aplica a operação seleção s_C, em que C é a condição (Campanhia aérea = Nadir) $\vee$ (Destino = Denver), ao banco de dados da Tabela 8?
14. O que você obtém quando aplica a projeção $P_{2,3,5}$ à 5-upla (a, b, c, d, e)?
15. Que operação projeção é usada para remover a primeira, a segunda e a quarta componentes de uma 6-upla?
16. Mostre a tabela obtida quando se aplica a projeção $P_{1,2,4}$ à Tabela 8.
17. Mostre a tabela obtida quando se aplica a projeção $P_{1,4}$ à Tabela 8.
18. Quantas componentes existem nas n-uplas na tabela obtida pela aplicação do operador junção (ou conjunção) J_3 a duas tabelas com 5-uplas e 8-uplas, respectivamente?
19. Construa a tabela obtida pela aplicação do operador (ou conjunção) junção J_2 às relações nas tabelas 9 e 10.
20. Mostre que se C_1 e C_2 forem condições às quais os elementos da relação n-ária R possam satisfazer, então $s_{C_1 \wedge C_2}(R) = s_{C_1}(s_{C_2}(R))$.
21. Mostre que se C_1 e C_2 forem condições às quais os elementos da relação n-ária R possam satisfazer, então $s_{C_1}(s_{C_2}(R)) = s_{C_2}(s_{C_1}(R))$.
22. Mostre que se C for uma condição à qual os elementos das relações n-árias R e S possam satisfazer, então $s_C(R \cup S) = s_C(R) \cup s_C(S)$.
23. Mostre que se C for uma condição à qual os elementos das relações n-árias R e S possam satisfazer, então $s_C(R \cap S) = s_C(R) \cap s_C(S)$.

TABELA 9 Partes_necessarias.

Fornecedor	*Numero_da_parte*	*Projeto*
23	1092	1
23	1101	3
23	9048	4
31	4975	3
31	3477	2
32	6984	4
32	9191	2
33	1001	1

TABELA 10 Partes_em_estoque.

Numero_da_parte	*Projeto*	*Quantidade*	*Codigo_de_cor*
1001	1	14	8
1092	1	2	2
1101	3	1	1
3477	2	25	2
4975	3	6	2
6984	4	10	1
9048	4	12	2
9191	2	80	4

24. Mostre que se C for uma condição à qual os elementos das relações n-árias R e S possam satisfazer, então $s_C(R - S) = s_C(R) - s_C(S)$.

25. Mostre que se R e S forem ambas relações n-árias, então $P_{i_1, i_2, \ldots, i_m}(R \cup S) = P_{i_1, i_2, \ldots, i_m}(R) \cup P_{i_1, i_2, \ldots, i_m}(S)$.

26. Dê um exemplo para mostrar que se R e S forem ambas relações n-árias, então $P_{i_1, i_2, \ldots, i_m}(R \cap S)$ pode ser diferente de $P_{i_1, i_2, \ldots, i_m}(R) \cap P_{i_1, i_2, \ldots, i_m}(S)$.

27. Dê um exemplo para mostrar que se R e S forem ambas relações n-árias, então $P_{i_1, i_2, \ldots, i_m}(R - S)$ pode ser diferente de $P_{i_1, i_2, \ldots, i_m}(R) - P_{i_1, i_2, \ldots, i_m}(S)$.

28. **a)** Quais são as operações que correspondem à consulta expressa usando este comando SQL?

```
SELECT Fornecedor
FROM Partes_necessarias
WHERE 1000 ≤ Numero_da_parte ≤ 5000
```

b) Qual é a saída desta consulta, tendo o banco de dados da Tabela 9 como entrada?

29. **a)** Quais são as operações que correspondem à consulta expressa usando este comando SQL?

```
SELECT Fornecedor,      Projeto
FROM Partes_            Partes_em_
necessarias,            estoque
WHERE Quantidade ≤ 10
```

b) Qual é a saída desta consulta, tendo os bancos de dados das tabelas 9 e 10 como entrada?

30. Determine se existe uma chave primária para a relação no Exemplo 2.

31. Determine se existe uma chave primária para a relação no Exemplo 3.

32. Mostre que uma relação n-ária com uma chave primária pode ser pensada como o gráfico de uma função que leva valores da chave primária a $(n - 1)$-uplas formadas a partir de valores nos outros domínios.

8.3 Representação de Relações

Introdução

Nesta seção, e no restante deste capítulo, todas as relações que estudaremos serão relações binárias. Por causa disso, nesta seção e no resto do capítulo a palavra relação se referirá sempre a uma relação binária. Há muitas formas de representar uma relação entre conjuntos finitos. Como vimos na Seção 8.1, uma maneira é listar todos os pares ordenados. Outra maneira de representar uma relação é usar uma tabela, como fizemos no Exemplo 3 da Seção 8.1. Nesta seção, discutiremos dois métodos alternativos para representar relações. Um método usa matrizes zero-um. O outro método usa representações pictóricas chamadas de grafos orientados, que discutiremos mais adiante nesta seção.

Em geral, as matrizes são apropriadas para a representação de relações em programas computacionais. Por outro lado, as pessoas em geral acham a representação de relações usando grafos orientados úteis para entender as propriedades dessas relações.

Representação de Relações Usando Matrizes

Uma relação entre conjuntos finitos pode ser representada usando uma matriz zero-um. Suponha que R seja uma relação de $A = \{a_1, a_2, \ldots, a_m\}$ em $B = \{b_1, b_2, \ldots, b_n\}$. (Aqui os elementos dos conjuntos A e B foram listados em uma ordem particular, mas arbitrária. Além disso, quando $A = B$, usamos a mesma ordem para A e B.) A relação R pode ser representada pela matriz $\mathbf{M}_R = [m_{ij}]$, em que

$$m_{ij} = \begin{cases} 1 \text{ se } (a_i, b_j) \in R, \\ 0 \text{ se } (a_i, b_j) \notin R. \end{cases}$$

Em outras palavras, a matriz zero-um que representa R tem um 1 como seu elemento (i, j) quando a_i está relacionado a b_j, e um 0 nesta posição se a_i não estiver relacionado a b_j. (Essa representação depende da ordem usada para A e B.)

O uso de matrizes para representar relações é ilustrado nos exemplos 1 a 6.

EXEMPLO 1 Suponha que $A = \{1, 2, 3\}$ e $B = \{1, 2\}$. Seja R a relação de A para B que contém (a, b) se $a \in A$, $b \in B$ e $a > b$. Qual é a matriz que representa R se $a_1 = 1$, $a_2 = 2$ e $a_3 = 3$, e $b_1 = 1$ e $b_2 = 2$?

Solução: Como $R = \{(2, 1), (3, 1), (3, 2)\}$, a matriz para R é

$$\mathbf{M}_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Os 1s em $\mathbf{M}_R$ mostram que os pares (2, 1), (3, 1) e (3, 2) pertencem a R. Os 0s mostram que nenhum outro par pertence a R. ◀

EXEMPLO 2 Sejam $A = \{a_1, a_2, a_3\}$ e B = $\{b_1, b_2, b_3, b_4, b_5\}$. Quais pares ordenados estão na relação R, representada pela matriz

$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}?$$

Solução: Como R consiste nos pares ordenados (a_i, b_j) com $m_{ij} = 1$, segue que

$$R = \{(a_1, b_2), (a_2, b_1), (a_2, b_3), (a_2, b_4), (a_3, b_1), (a_3, b_3), (a_3, b_5)\}.$$ ◀

A matriz de uma relação em um conjunto, que é uma matriz quadrada, pode ser usada para determinar se a relação tem determinadas propriedades. Lembre-se de que uma relação R em A é reflexiva se $(a, a) \in R$ sempre que $a \in A$. Assim, R é reflexiva se e somente se $(a_i, a_i) \in R$ para $i = 1, 2, \ldots, n$. Portanto, R é reflexiva se e somente se $m_{ii} = 1$, para $i = 1, 2, \ldots, n$. Em outras palavras, R é reflexiva se todos os elementos na diagonal principal de $\mathbf{M}_R$ forem iguais a 1, como mostra a Figura 1.

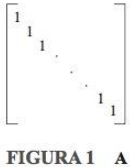
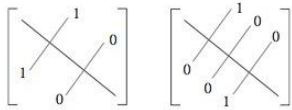


FIGURA 1 A Matriz Zero-Um para uma Relação Reflexiva.

A relação R é simétrica se $(a, b) \in R$ implica que $(b, a) \in R$. Conseqüentemente, a relação R em um conjunto $A = \{a_1, a_2, \ldots, a_n\}$ é simétrica se e somente se $(a_j, a_i) \in R$ sempre que $(a_i, a_j) \in R$. Em termos dos elementos de $\mathbf{M}_R$, R é simétrica se e somente se $m_{ji} = 1$ sempre que $m_{ij} = 1$. Isso também significa que $m_{ji} = 0$ sempre que $m_{ij} = 0$. Conseqüentemente, R é simétrica se e somente se $m_{ij} = m_{ji}$, para todos os pares de inteiros i e j, com $i = 1, 2, \ldots, n$ e $j = 1, 2, \ldots, n$. Lembrando a definição de transposição de uma matriz da Seção 3.8, vemos que R é simétrica se e somente se

$$\mathbf{M}_R = (\mathbf{M}_R)^t,$$



(a) Simétrica (b) Anti-simétrica

FIGURA 2 As Matrizes Zero-Um para Relações Simétricas e Anti-simétricas

isto é, se $\mathbf{M}_R$ for uma matriz simétrica. A forma da matriz para uma relação simétrica está ilustrada na Figura 2(a).

A relação R é anti-simétrica se e somente se $(a, b) \in R$ e $(b, a) \in R$ implica que $a = b$. Conseqüentemente, a matriz de uma relação anti-simétrica tem a propriedade que se $m_{ij} = 1$ com $i \neq j$, então $m_{ji} = 0$. Ou, em outras palavras, ou $m_{ij} = 0$ ou $m_{ji} = 0$ quando $i \neq j$. A forma da matriz para uma relação anti-simétrica está ilustrada na Figura 2(b).

EXEMPLO 3 Suponha que a relação R em um conjunto seja representada pela matriz

$$\mathbf{M}_R = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

R é reflexiva, simétrica e/ou anti-simétrica?

Solução: Como todos os elementos da diagonal dessa matriz são iguais a 1, R é reflexiva. Além disso, como $\mathbf{M}_R$ é simétrica, segue que R é simétrica. Também é fácil de ver que R não é anti-simétrica. ◀

As operações booleanas conjunção e encontro (discutidas na Seção 3.8) podem ser usadas para encontrar as matrizes que representam a união e a intersecção de duas relações. Suponha que R_1 e R_2 sejam relações em um conjunto A representadas pelas matrizes $\mathbf{M}_{R_1}$ e $\mathbf{M}_{R_2}$, respectivamente. A matriz que representa a união dessas relações tem um 1 nas posições nas quais ou $\mathbf{M}_{R_1}$ ou $\mathbf{M}_{R_2}$ tem um 1. A matriz que representa a intersecção dessas relações tem um 1 nas posições em que ambas, $\mathbf{M}_{R_1}$ e $\mathbf{M}_{R_2}$, têm um 1. Assim, as matrizes que representam a união e a intersecção dessas relações são

$$\mathbf{M}_{R_1 \cup R_2} = \mathbf{M}_{R_1} \vee \mathbf{M}_{R_2} \quad \text{e} \quad \mathbf{M}_{R_1 \cap R_2} = \mathbf{M}_{R_1} \wedge \mathbf{M}_{R_2}.$$

EXEMPLO 4 Suponha que as relações R_1 e R_2 em um conjunto A sejam representadas pelas matrizes

$$\mathbf{M}_{R1} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{e} \quad \mathbf{M}_{R2} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Quais são as matrizes que representam $R_1 \cup R_2$ e $R_1 \cap R_2$?

Solução: As matrizes dessas relações são

$$\mathbf{M}_{R_1 \cup R_2} = \mathbf{M}_{R_1} \vee \mathbf{M}_{R_2} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix},$$

$$\mathbf{M}_{R_1 \cap R_2} = \mathbf{M}_{R_1} \wedge \mathbf{M}_{R_2} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

◀

Agora, voltamos nossa atenção para determinar a matriz para a composta de relações. Essa matriz pode ser encontrada usando o produto booleano das matrizes (discutido na Seção 3.8) para essas relações. Em particular, suponha que R seja uma relação de A em B e S seja uma relação de B em C. Suponha que A, B e C tenham m, n e p elementos, respectivamente. Sejam $S \circ R$, R e S as matrizes zero-um para $\mathbf{M}_{S \circ R} = [t_{ij}]$, $\mathbf{M}_R = [r_{ij}]$ e $\mathbf{M}_S = [s_{ij}]$, respectivamente (estas matrizes têm tamanhos $m \times p$, $m \times n$ e $n \times p$, respectivamente). O par ordenado (a_i, c_j) pertence a $S \circ R$ se e somente se existir um elemento b_k, tal que (a_i, b_k) pertença a R e (b_k, c_j) pertença a S. Segue que $t_{ij} = 1$ se e somente se $r_{ik} = s_{kj} = 1$ para algum k. Da definição de produto booleano, segue que

$$\mathbf{M}_{S \circ R} = \mathbf{M}_R \odot \mathbf{M}_S.$$

EXEMPLO 5 Encontre a matriz que representa $S \circ R$, quando as matrizes que representam R e S são

$$\mathbf{M}_R = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{e} \quad \mathbf{M}_S = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

Solução: A matriz para $S \circ R$ é

$$\mathbf{M}_{S \circ R} = \mathbf{M}_R \odot \mathbf{M}_S = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

◀

A matriz que representa a composta de duas relações pode ser usada para encontrar a matriz para $\mathbf{M}_{R^n}$. Em particular,

$$\mathbf{M}_{R^n} = \mathrm{M}_R^{[n]},$$

a partir da definição de potências booleanas. O Exercício 35, no final desta seção, pede uma demonstração dessa fórmula.

EXEMPLO 6 Encontre a matriz que representa R^2, quando a matriz que representa R é

$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Solução: A matriz para R_2 é

$$\mathbf{M}_{R^2} = \mathbf{M}_R^{[2]} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

◀

Representação de Relações Usando Dígrafos

Mostramos que uma relação pode ser representada listando todos os seus pares ordenados ou usando uma matriz zero-um. Existe outra maneira importante de representar uma relação usando uma representação pictórica. Cada elemento do conjunto é representado por um ponto e cada par ordenado é representado usando um arco com sua direção indicada por uma flecha. Usamos essa representação pictórica quando pensamos em relações, em um conjunto finito, como **grafos orientados** ou **dígrafos**.

DEFINIÇÃO 1 Um *grafo orientado*, ou *dígrafo*, consiste em um conjunto V de *vértices* (ou *nós*) junto com um conjunto E de pares ordenados de elementos de V chamados de *arestas* (ou *arcos*). O vértice a é chamado de *vértice inicial* de uma aresta (a, b) e o vértice b é chamado de *vértice final* da aresta.

Uma aresta da forma (a, a) é representada usando um arco do vértice a que volte para ele mesmo. Essa aresta é chamada de **laço**.

EXEMPLO 7 O grafo orientado com vértices a, b, c e d e arestas (a, b), (a, d), (b, b), (b, d), (c, a), (c, b) e (d, b) é mostrado na Figura 3. ◀

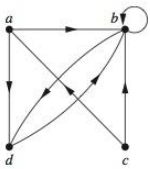


FIGURA 3
Um Grafo Orientado.

A relação R em um conjunto A é representada por um grafo orientado que tem os elementos de A como seus vértices e os pares ordenados (a, b), em que $(a, b) \in R$, como arestas. Essa associação se constitui em uma correspondência biunívoca entre as relações em um conjunto A e os grafos orientados com A como seu conjunto de vértices. Assim, toda afirmação sobre relações corresponde a uma afirmação sobre grafos orientados e vice-versa. Grafos orientados dão uma representação visual da informação sobre relações. Por esse motivo, são freqüentemente usados para estudar relações e suas propriedades. (Observe que as relações de um conjunto A em um conjunto B podem ser representadas por um grafo orientado no qual existe um vértice para cada elemento de A e um vértice para cada elemento de B, como mostrado na Seção 8.1. Entretanto, quando $A = B$, essa representação fornece muito menos percepção do que as representações por dígrafos descritas aqui.) O uso de grafos orientados para representar relações em conjuntos é ilustrado nos exemplos 8 a 10.

EXEMPLO 8 O grafo orientado da relação

$$R = \{(1, 1), (1, 3), (2, 1), (2, 3), (2, 4), (3, 1), (3, 2), (4, 1)\}$$

no conjunto $\{1, 2, 3, 4\}$ é mostrado na Figura 4. ◀

EXEMPLO 9 Quais são os pares ordenados na relação R representada pelo grafo orientado mostrado na Figura 5?

Solução: Os pares ordenados (x, y) na relação são

$$R = \{(1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (3, 1), (3, 3), (4, 1), (4, 3)\}.$$

Cada um desses pares corresponde a uma aresta do grafo orientado, com (2, 2) e (3, 3) correspondendo aos laços. ◀

O grafo orientado que representa uma relação pode ser usado para determinar se a relação tem diversas propriedades. Por exemplo, uma relação é reflexiva se e somente se existir um laço em todo vértice do grafo orientado, de modo que todo par ordenado da forma (x, x) ocorra na relação. Uma relação é simétrica se e somente se para cada aresta entre vértices distintos em seu

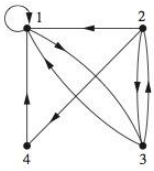


FIGURA 4 O Grafo Orientado da Relação *R*.

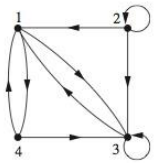
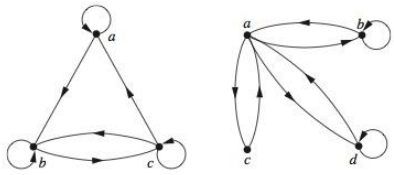


FIGURA 5 O Grafo Orientado da Relação *R*.



(a) Grafo orientado de *R* (b) Grafo orientado de *S*

FIGURA 6 Os Grafos Orientados das Relações *R* e *S*.

dígrafo existir uma aresta no sentido oposto, de modo que (y, x) esteja na relação sempre que (x, y) estiver na relação. Analogamente, uma relação é anti-simétrica se e somente se nunca existir duas arestas em sentidos opostos entre dois vértices distintos. Finalmente, uma relação é transitiva se e somente se sempre que existir uma aresta de um vértice x para um vértice y e uma aresta de um vértice y para um vértice z, existir uma aresta de x para z (completando um triângulo em que cada lado é uma aresta orientada com a direção correta).

Lembre-se: Uma relação simétrica pode ser representada por um grafo não orientado, que é um grafo cujas arestas não têm sentidos. Estudaremos grafos não orientados no Capítulo 9.

EXEMPLO 10 Determine se as relações para os grafos orientados mostrados na Figura 6 são reflexivas, simétricas, anti-simétricas e/ou transitivas.

Solução: Como existem laços em todos os vértices do grafo orientado para R, ela é reflexiva. R não é nem simétrica nem anti-simétrica, pois existe uma aresta de a para b, mas não uma de b para a, mas existem arestas, em ambos os sentidos, que ligam b e c. Finalmente, R não é transitiva, pois existe uma aresta de a para b e uma aresta de b para c, mas não uma aresta de a para c.

Como os laços não estão presentes em todos os vértices do grafo orientado para S, esta relação não é reflexiva. Ela é simétrica, mas não é anti-simétrica, pois toda aresta entre vértices distintos está acompanhada de uma aresta em sentido oposto. Também não é difícil de ver a partir do grafo orientado que S não é transitiva, pois (c, a) e (a, b) pertencem a S, mas (c, b) não pertence a S. ◀

Exercícios

1. Represente cada uma destas relações em {1, 2, 3} por uma matriz (com os elementos deste conjunto listados em ordem crescente).
- **a)** {(1, 1), (1, 2), (1, 3)}
- **b)** {(1, 2), (2, 1), (2, 2), (3, 3)}
- **c)** {(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)}
- **d)** {(1, 3), (3, 1)}

2. Represente cada uma destas relações em {1, 2, 3, 4} por uma matriz (com os elementos deste conjunto listados em ordem crescente).
- **a)** {(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)}
- **b)** {(1, 1), (1, 4), (2, 2), (3, 3), (4, 1)}
- **c)** {(1, 2), (1, 3), (1, 4), (2, 1), (2, 3), (2, 4), (3, 1), (3, 2), (3, 4), (4, 1), (4, 2), (4, 3)}
- **d)** {(2, 4), (3, 1), (3, 2), (3, 4)}

3. Liste os pares ordenados nas relações em {1, 2, 3} correspondentes a estas matrizes (em que linhas e colunas correspondem aos inteiros listados em ordem crescente).

a) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

4. Liste os pares ordenados nas relações em {1, 2, 3, 4} correspondentes a estas matrizes (em que linhas e colunas correspondem aos inteiros listados em ordem crescente).

a) $\begin{bmatrix}1&1&0&1\\1&0&1&0\\0&1&1&1\\1&0&1&1\end{bmatrix}$ **b)** $\begin{bmatrix}1&1&1&0\\0&1&0&0\\0&0&1&1\\1&0&0&1\end{bmatrix}$

c) $\begin{bmatrix}0&1&0&1\\1&0&1&0\\0&1&0&1\\1&0&1&0\end{bmatrix}$

5. Como a matriz que representa uma relação R em um conjunto A pode ser usada para determinar se a relação é irreflexiva?

6. Como a matriz que representa uma relação R em um conjunto A pode ser usada para determinar se a relação é assimétrica?

7. Determine se as relações representadas pelas matrizes no Exercício 3 são reflexivas, irreflexivas, simétricas, anti-simétricas e/ou transitivas.

8. Determine se as relações representadas pelas matrizes no Exercício 4 são reflexivas, irreflexivas, simétricas, anti-simétricas e/ou transitivas.

9. Quantos elementos não nulos a matriz que representa a relação R em $A = \{1, 2, 3, \ldots, 100\}$, que consiste nos primeiros 100 inteiros positivos, tem se R for

a) $\{(a, b) \mid a > b\}$? **b)** $\{(a, b) \mid a \neq b\}$?

c) $\{(a, b) \mid a = b + 1\}$? **d)** $\{(a, b) \mid a = 1\}$?

e) $\{(a, b) \mid ab = 1\}$?

10. Quantos elementos não nulos a matriz que representa a relação R em $A = \{1, 2, 3, \ldots, 1\,000\}$, que consiste nos primeiros 1.000 inteiros positivos, tem se R for

a) $\{(a, b) \mid a \leq b\}$?

b) $\{(a, b) \mid a = b \pm 1\}$?

c) $\{(a, b) \mid a + b = 1\,000\}$?

d) $\{(a, b) \mid a + b \leq 1\,001\}$?

e) $\{(a, b) \mid a \neq 0\}$?

11. Como pode ser encontrada a matriz para $\overline{R}$, o complementar da relação R, a partir da matriz que representa R, quando R é uma relação em um conjunto finito A?

12. Como pode ser encontrada a matriz para R^{-1}, a inversa da relação R, a partir da matriz que representa R, quando R é uma relação em um conjunto finito A?

13. Seja R a relação representada pela matriz

$$\mathbf{M}_R = \begin{bmatrix}0&1&1\\1&1&0\\1&0&1\end{bmatrix}.$$

Encontre a matriz que representa

a) R^{-1}. **b)** $\overline{R}$. **c)** R^2.

14. Sejam R_1 e R_2 relações em um conjunto A representadas pelas matrizes

$$\mathbf{M}_{R_1} = \begin{bmatrix}0&1&0\\1&1&1\\1&0&0\end{bmatrix} \quad \text{e} \quad \mathbf{M}_{R_2} = \begin{bmatrix}0&1&0\\0&1&1\\1&1&1\end{bmatrix}.$$

Encontre as matrizes que representam

a) $R_1 \cup R_2$. **b)** $R_1 \cap R_2$. **c)** $R_2 \circ R_1$.

d) $R_1 \circ R_1$. **e)** $R_1 \oplus R_2$.

15. Seja R a relação representada pela matriz

$$\mathbf{M}_R = \begin{bmatrix}0&1&0\\0&0&1\\1&1&0\end{bmatrix}.$$

Encontre as matrizes que representam

a) R^2. **b)** R^3. **c)** R^4.

16. Seja R uma relação em um conjunto A com n elementos. Se existirem k elementos não nulos em $\mathbf{M}_R$, a matriz que representa R, quantos elementos não nulos existem em $\mathbf{M}_{R^{-1}}$, a matriz que representa R^{-1}, a inversa de R?

17. Seja R uma relação em um conjunto A com n elementos. Se existirem k elementos não nulos em $\mathbf{M}_R$, a matriz que representa R, quantos elementos não nulos existem em $\mathbf{M}_{\overline{R}}$, a matriz que representa $\overline{R}$, o complemento de R?

18. Trace os grafos orientados que representam cada uma das relações do Exercício 1.

19. Trace os grafos orientados que representam cada uma das relações do Exercício 2.

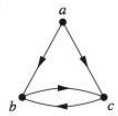
20. Trace os grafos orientados que representam cada uma das relações do Exercício 3.

21. Trace os grafos orientados que representam cada uma das relações do Exercício 4.

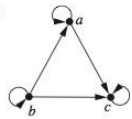
22. Trace o grafo orientado que representa a relação $\{(a, a), (a, b), (b, c), (c, b), (c, d), (d, a), (d, b)\}$.

Nos exercícios 23 a 28, liste os pares ordenados nas relações representadas pelos grafos orientados.

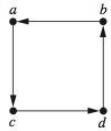
23.



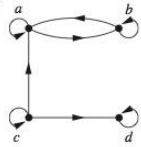
24.



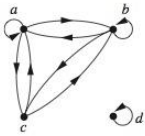
25.



26.



27.



28.

29. Como o grafo orientado de uma relação R em um conjunto finito A pode ser usado para determinar se uma relação é assimétrica?

30. Como o grafo orientado de uma relação R em um conjunto finito A pode ser usado para determinar se uma relação é irreflexiva?

31. Determine se as relações representadas pelos grafos orientados mostrados nos exercícios 23 a 25 são reflexivas, irreflexivas, simétricas, anti-simétricas e/ou transitivas.

32. Determine se as relações representadas pelos grafos orientados mostrados nos exercícios 26 a 28 são reflexivas, irreflexivas, simétricas, anti-simétricas, assimétricas e/ou transitivas.

33. Seja R uma relação em um conjunto A. Explique como usar o grafo orientado que representa R para obter o grafo orientado que representa a relação inversa R^{-1}.

34. Seja R uma relação em um conjunto A. Explique como usar o grafo orientado que representa R para obter o grafo orientado que representa a relação complementar $\overline{R}$.

35. Mostre que se $\mathbf{M}_R$ for a matriz que representa a relação R, então $\mathbf{M}_R^{[n]}$ é a matriz que representa a relação R^n.

36. Dados os grafos orientados que representam duas relações, como podem ser encontrados os grafos orientados da união, da intersecção, da diferença simétrica, da diferença e da composição dessas relações?

8.4 Fecho de Relações

Introdução

Uma rede de computadores tem centro de dados em Boston, Chicago, Denver, Detroit, Nova York e San Diego. Existem linhas telefônicas diretas, de mão única, de Boston a Chicago, de Boston a Detroit, de Chicago a Detroit, de Detroit a Denver e de Nova York a San Diego. Seja R a relação que contém (a, b) se existir uma linha telefônica a partir do centro de dados em a para aquele em b. Como podemos determinar se existe alguma ligação (possivelmente indireta) composta de uma ou mais linhas telefônicas de um centro para outro? Como nem todas as ligações são diretas, como a ligação de Boston a Denver que passa por Detroit, R não pode ser usada diretamente para responder isso. Na linguagem de relações, R não é transitiva, de modo que ela não contém todos os pares que podem ser ligados. Como mostraremos nesta seção, podemos encontrar todos os pares de centro de dados que têm uma ligação construindo uma relação transitiva S que contenha R, tal que S seja um subconjunto de qualquer relação transitiva que contenha R. Aqui, S é a menor relação transitiva que contém R. Esta relação é chamada de **fecho transitivo** de R.

Em geral, seja R uma relação em um conjunto A. R pode ou não ter alguma propriedade **P**, tal como refletividade, simetria ou transitividade. Se existir uma relação S com a propriedade **P** que contenha R, tal que S seja um subconjunto de toda a relação com a propriedade **P** que contenha R, então S é chamada de **fecho** de R relativo a **P**. (Observe que o fecho de uma relação relativo a uma propriedade pode não existir; veja os exercícios 15 e 35 no final desta seção.) Mostraremos como os fechos reflexivo, simétrico e transitivo de relações podem ser encontrados.

Fechos

A relação $R = \{(1, 1), (1, 2), (2, 1), (3, 2)\}$ no conjunto $A = \{1, 2, 3\}$ não é reflexiva. Como podemos produzir uma relação reflexiva que contenha R, que seja a menor possível? Isso pode ser feito acrescentando $(2, 2)$ e $(3, 3)$ a R, pois estes são os únicos pares da forma (a, a) que não estão em R. Claramente, esta nova relação contém R. Além disso, *qualquer* relação reflexiva que contenha R deve também conter $(2, 2)$ e $(3, 3)$. Como esta relação contém R, é reflexiva e está contida em qualquer relação reflexiva que contenha R, ela é chamada de **fecho reflexivo** de R.

Como este exemplo ilustra, dada uma relação R em um conjunto A, o fecho reflexivo de R pode ser formado acrescentando a R todos os pares da forma (a, a), com $a \in A$, que ainda não

estejam em R. A adição desses pares produz uma nova relação que é reflexiva, contém R e está contida em qualquer relação reflexiva que contenha R. Vemos que o fecho reflexivo de R é igual a $R \cup \Delta$, em que $\Delta = \{(a, a) \mid a \in A\}$ é a **relação diagonal** em A. (O leitor deve verificar isso.)

EXEMPLO 1 Qual é o fecho reflexivo da relação $R = \{(a, b) \mid a < b\}$ no conjunto dos inteiros?

Solução: O fecho reflexivo de R é

$$R \cup \Delta = \{(a, b) \mid a < b\} \cup \{(a, a) \mid a \in \mathbf{Z}\} = \{(a, b) \mid a \leq b\}. \quad \blacktriangleleft$$

A relação $\{(1, 1), (1, 2), (2, 2), (2, 3), (3, 1), (3, 2)\}$ em $\{1, 2, 3\}$ não é simétrica. Como podemos produzir uma relação simétrica que seja tão pequena quanto possível e que contenha R? Para fazer isso, precisamos apenas acrescentar $(2, 1)$ e $(1, 3)$, pois estes são os únicos pares da forma (b, a) com $(a, b) \in R$ que não estão em R. Esta nova relação é simétrica e contém R. Além disso, *qualquer* relação simétrica que contenha R deve conter $(2, 1)$ e $(1, 3)$. Conseqüentemente, esta nova relação é chamada de **fecho simétrico** de R.

Como este exemplo ilustra, o fecho simétrico de uma relação R pode ser construído adicionando-se todos os pares ordenados da forma (b, a), em que (a, b) está na relação, que ainda não estejam presentes em R. A adição desses pares produz uma relação que é simétrica, que contém R e que está contida em qualquer relação simétrica que contenha R. O fecho simétrico de uma relação pode ser construído tomando-se a união de uma relação com sua inversa (definida no preâmbulo do Exercício 24 da Seção 8.1); ou seja, $R \cup R^{-1}$ é o fecho simétrico de R, em que $R^{-1} = \{(b, a) \mid (a, b) \in R\}$. O leitor deve verificar esta afirmação.

EXEMPLO 2 Qual é o fecho simétrico da relação $R = \{(a, b) \mid a > b\}$ no conjunto dos inteiros positivos?

Exemplos Extras

Solução: O fecho simétrico de R é a relação

$$R \cup R^{-1} = \{(a, b) \mid a > b\} \cup \{(b, a) \mid a > b\} = \{(a, b) \mid a \neq b\}.$$

Esta última igualdade segue do fato de R conter todos os pares ordenados de inteiros positivos nos quais o primeiro elemento é maior do que o segundo elemento e R^{-1} conter todos os pares ordenados de inteiros positivos nos quais o primeiro elemento é menor do que o segundo. ◀

Suponha que uma relação R não seja transitiva. Como podemos produzir uma relação transitiva que contenha R, tal que esta nova relação esteja contida em qualquer relação transitiva que contenha R? O fecho transitivo de uma relação R pode ser obtido acrescentando-se todos os pares da forma (a, c), em que (a, b) e (b, c) já estão na relação? Considere a relação $R = \{(1, 3), (1, 4), (2, 1), (3, 2)\}$ no conjunto $\{1, 2, 3, 4\}$. Esta relação não é transitiva, pois não contém todos os pares da forma (a, c), em que (a, b) e (b, c) estejam em R. Os pares desta forma que não estão em R são $(1, 2), (2, 3), (2, 4)$ e $(3, 1)$. Acrescentar estes pares *não* produz uma relação transitiva, pois a relação resultante contém $(3, 1)$ e $(1, 4)$, mas não contém $(3, 4)$. Isso mostra que construir o fecho transitivo de uma relação é mais complicado que construir tanto o fecho reflexivo quanto o simétrico. O resto desta seção desenvolve algoritmos para a construção de fechos transitivos. Como será mostrado mais tarde nesta seção, o fecho transitivo de uma relação pode ser encontrado acrescentando-se novos pares ordenados que devam estar presentes e repetindo-se esse processo até que nenhum novo par ordenado seja necessário.

Caminhos em Grafos Orientados

Veremos que representar relações por grafos orientados ajuda na construção de fechos transitivos. Introduziremos agora alguma terminologia que usaremos para esse propósito.

Um caminho em um grafo orientado é obtido percorrendo arestas (na mesma direção indicada pela flecha na aresta).

DEFINIÇÃO 1 Um *caminho* de a para b no grafo orientado G é uma seqüência de arestas (x_0, x_1), (x_1, x_2), (x_2, x_3), …, (x_{n-1}, x_n) em G, em que n é um inteiro não negativo e $x_0 = a$ e $x_n = b$, ou seja, uma seqüência de arestas na qual o vértice final de uma aresta é o mesmo que o vértice inicial da próxima aresta do caminho. Este caminho é indicado por $x_0, x_1, x_2, \ldots, x_{n-1}, x_n$ e tem *comprimento n*. Podemos considerar um conjunto vazio de arestas como um caminho de a para a. Um caminho de comprimento $n \geq 1$ que começa e termina em um mesmo vértice é chamado de *circuito* ou *ciclo*.

Um caminho em um grafo orientado pode passar por um vértice mais de uma vez. Além disso, uma aresta em um grafo orientado pode ocorrer mais de uma vez em um caminho.

EXEMPLO 3 Quais dos seguintes são caminhos no grafo orientado mostrado na Figura 1: a, b, e, d; a, e, c, d, b; b, a, c, b, a, a, b; d, c; c, b, a; e, b, a, b, a, b, e? Quais são os comprimentos daqueles que forem caminhos? Quais dos caminhos da lista são circuitos?

Solução: Como cada um de (a, b), (b, e) e (e, d) são arestas, a, b, e, d é um caminho de comprimento três. Como (c, d) não é uma aresta, a, e, c, d, b não é um caminho. Além disso, b, a, c, b, a, a, b é um caminho de comprimento seis, pois (b, a), (a, c), (c, b), (b, a), (a, a) e (a, b) são todos arestas. Vemos que d, c é um caminho de comprimento um, pois (d, c) é uma aresta. Além disso, c, b, a é um caminho de comprimento dois, pois (c, b) e (b, a) são arestas. Todos (e, b), (b, a), (a, b), (b, a), (a, b) e (b, e) são arestas, de modo que e, b, a, b, a, b, e é um caminho de comprimento seis.

Os dois caminhos b, a, c, b, a, a, b e e, b, a, b, a, b, e são circuitos, pois eles começam e terminam no mesmo vértice. Os caminhos a, b, e, d; c, b, a e d, c não são circuitos. ◀

O termo *caminho* também se aplica a relações. Levando a definição de grafos orientados para relações, existe um **caminho** de a para b em R se existir uma seqüência de elementos $a, x_1, x_2, \ldots, x_{n-1}, b$, com $(a, x_1) \in R$, $(x_1, x_2) \in R$, …, e $(x_{n-1}, b) \in R$. O Teorema 1 pode ser obtido da definição de um caminho em uma relação.

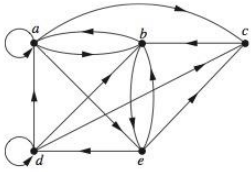


FIGURA 1 Um Grafo Orientado.

TEOREMA 1 Seja R uma relação em um conjunto A. Existe um caminho de comprimento n, em que n é um inteiro positivo, de a a b se e somente se $(a, b) \in R^n$.

Demonstração: Usaremos indução matemática. Por definição, existe um caminho de a a b de comprimento 1 se e somente se $(a, b) \in R$, de modo que o teorema é verdadeiro quando $n = 1$.

Suponha que o teorema seja verdadeiro para o inteiro positivo n. Esta é a hipótese de indução. Existe um caminho de comprimento $n + 1$ de a para b se e somente se existir um elemento $c \in A$, tal que exista um caminho de comprimento um de a para c, de modo que $(a, c) \in R$, e um caminho de comprimento n de c para b, isto é, $(a, c) \in R^n$. Conseqüentemente, pela hipótese de indução, existe um caminho de comprimento $n + 1$ de a a b se e somente se existir um elemento c com $(a, c) \in R$ e $(c, b) \in R^n$. No entanto, existe esse elemento se e somente se $(a, b) \in R^{n+1}$. Portanto, existe um caminho de comprimento $n + 1$ de a a b se e somente se $(a, b) \in R^{n+1}$. Isto completa a demonstração. ◁

Fechos Transitivos

Mostramos agora que encontrar o fecho transitivo de uma relação é equivalente a determinar quais pares de vértices no grafo orientado associado são ligados por um caminho. Com isso em mente, definimos uma nova relação.

DEFINIÇÃO 2 Seja R uma relação em um conjunto A. A *relação de conectividade* R^* consiste nos pares (a, b), tal que existe um caminho de comprimento pelo menos um, de a a b em R.

Como R^n consiste nos pares (a, b), tal que existe um caminho de comprimento n de a para b, segue que R^* é a união de todos os conjuntos R^n. Em outras palavras,

$$R^* = \bigcup_{n=1}^{\infty} R^n.$$

A relação de conectividade é útil em muitos modelos.

EXEMPLO 4 Seja R a relação no conjunto de todas as pessoas do mundo que contém (a, b) se a já encontrou b. O que é R^n, quando n é um inteiro positivo maior do que um? O que é R^*?

Solução: A relação R^2 contém (a, b) se existir uma pessoa c, tal que $(a, c) \in R$ e $(c, b) \in R$, ou seja, se existir uma pessoa c, tal que a já encontrou c e c já encontrou b. Analogamente, R^n consiste em todos os pares (a, b), tal que existam pessoas $x_1, x_2, \ldots, x_{n-1}$, tal que a já encontrou x_1, x_1 já encontrou x_2, …, e x_{n-1} já encontrou b.

A relação R^* contém (a, b) se existir uma seqüência de pessoas, começando com a e terminando com b, tal que cada pessoa na seqüência já encontrou a próxima pessoa na seqüência. (Existem muitas conjecturas interessantes sobre R^*. Você acha que esta relação de conectividade inclui o par com você como primeiro elemento e o presidente da Mongólia como segundo elemento? Usaremos grafos para modelar esta aplicação no Capítulo 9.) ◀

EXEMPLO 5 Seja R a relação no conjunto de todas as paradas de metrô na cidade de Nova York que contém (a, b) se for possível viajar da parada a para a parada b sem mudar de trem. O que é R^n quando n é um inteiro positivo? O que é R^*?

Solução: A relação R^n contém (a, b) se for possível viajar da parada a para a parada b fazendo $n - 1$ mudanças de trens. A relação R^* consiste nos pares ordenados (a, b) para os quais seja possível viajar da parada a para a parada b fazendo tantas mudanças de trens quanto necessário. (O leitor deve verificar essas afirmações.) ◀

EXEMPLO 6 Seja R a relação no conjunto de todos os estados dos Estados Unidos que contém (a, b) se o estado a e o estado b tiverem uma fronteira comum. O que é R^n, em que n é um inteiro positivo? O que é R^*?

Solução: A relação R^n consiste nos pares (a, b), para os quais é possível ir do estado a para o estado b cruzando exatamente n fronteiras estaduais. R^* consiste nos pares ordenados (a, b), para os quais é possível ir do estado a para o estado b cruzando tantas fronteiras quanto necessário. (O leitor deve verificar essas afirmações.) Os únicos pares ordenados que não estão em R^* são aqueles que contêm estados que não estão conectados com os Estados Unidos continental (isto é, os pares que contêm o Alasca ou o Havaí). ◀

O Teorema 2 mostra que o fecho transitivo de uma relação e a relação de conectividade associada são iguais.

TEOREMA 2 O fecho transitivo de uma relação R é igual à relação de conectividade R^*.

Demonstração: Observe que R^* contém R por definição. Para mostrar que R^* é o fecho transitivo de R, devemos mostrar também que R^* é transitiva e que $R^* \subseteq S$ sempre que S for uma relação transitiva que contenha R.

Primeiro, mostramos que R^* é transitiva. Se $(a, b) \in R^*$ e $(b, c) \in R^*$, então existem caminhos de a para b e de b para c em R. Obtemos um caminho de a para a c começando com um caminho de a para a b e seguindo-o com um caminho de b para c. Logo, $(a, c) \in R^*$. Segue que R^* é transitiva.

Suponha agora que S seja uma relação transitiva que contenha R. Como S é transitiva, S^n também é transitiva (o leitor deve verificar isso) e $S^n \subseteq S$ (pelo Teorema 1 da Seção 8.1). Além disso, como

$$S^* = \bigcup_{k=1}^{\infty} S^k$$

e $S^k \subseteq S$, segue que $S^* \subseteq S$. Observe agora que se $R \subseteq S$, então $R^* \subseteq S^*$, pois qualquer caminho em R também é um caminho em S. Conseqüentemente, $R^* \subseteq S^* \subseteq S$. Assim, qualquer relação transitiva que contenha R também deve conter R^*. Portanto, R^* é o fecho transitivo de R. ◁

Agora que sabemos que o fecho transitivo é igual à relação de conectividade, voltamos nossa atenção ao problema de calcular essa relação. Não precisamos examinar caminhos arbitrariamente longos para determinar se existe um caminho entre dois vértices de um grafo orientado finito. Como mostra o Lema 1, é suficiente examinar caminhos que não contenham mais de n arestas, em que n é o número de elementos do conjunto.

LEMA 1 Sejam A um conjunto com n elementos e R uma relação em A. Se existir um caminho de comprimento pelo menos um, em R de a para b, então existirá um tal caminho com comprimento que não exceda n. Além disso, quando $a \neq b$, se existir um caminho de comprimento pelo menos um, em R de a para b, então existirá um tal caminho com comprimento que não exceda $n - 1$.

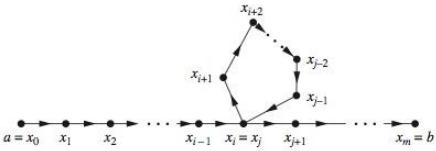


FIGURA 2 **Produzindo um Caminho com Comprimento que Não Exceda *n*.**

Demonstração: Suponha que existe um caminho de a para b em R. Seja m o comprimento do caminho mais curto. Suponha que $x_0, x_1, x_2, \ldots, x_{m-1}, x_m$, em que $x_0 = a$ e $x_m = b$, seja esse caminho.

Suponha que $a = b$ e que $m > n$, de modo que $m \geq n + 1$. Pelo princípio da casa dos pombos, como existem n vértices em A, entre os m vértices $x_0, x_1, \ldots, x_{m-1}$, pelo menos dois são iguais (ver a Figura 2).

Suponha que $x_i = x_j$, com $0 \leq i < j \leq m - 1$. Então, o caminho contém um circuito de x_i para si mesmo. Esse circuito pode ser removido do caminho de a para b, deixando um caminho, ou seja, $x_0, x_1, \ldots, x_i, x_{j+1}, \ldots, x_{m-1}, x_m$, de a para b de comprimento menor. Logo, o caminho de menor comprimento deve ter comprimento menor que ou igual a n.

O caso em que $a \neq b$ é deixado como um exercício para o leitor. ◁

Do Lema 1, vemos que o fecho transitivo de R é a união de R, R^2, R^3, ..., e R^n. Isso segue porque existe um caminho em R^* entre dois vértices se e somente se existir um caminho entre esses vértices em R^i, para algum inteiro positivo i, com $i \leq n$. Como

$$R^* = R \cup R^2 \cup R^3 \cup \cdots \cup R^n$$

e a matriz zero-um que representa uma união das relações é a conjunção das matrizes zero-um dessas relações, a matriz zero-um do fecho transitivo é a conjunção das matrizes zero-um das primeiras n potências da matriz zero-um de R.

TEOREMA 3 Seja $\mathbf{M}_R$ a matriz zero-um da relação R em um conjunto com n elementos. Então, a matriz zero-um do fecho transitivo R^* é

$$\mathbf{M}_{R^*} = \mathbf{M}_R \vee \mathbf{M}_R^{[2]} \vee \mathbf{M}_R^{[3]} \vee \cdots \vee \mathbf{M}_R^{[n]}.$$

EXEMPLO 7 Encontre a matriz zero-um do fecho transitivo da relação R em que

$$\mathbf{M}_R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

Solução: Do Teorema 3, segue que a matriz zero-um de R^* é

$$\mathbf{M}_{R^*} = \mathbf{M}_R \vee \mathbf{M}_R^{[2]} \vee \mathbf{M}_R^{[3]}.$$

Como

$$\mathbf{M}_R^{[2]} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{M}_R^{[3]} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix},$$

segue que

$$\mathbf{M}_{R^*} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

◀

Links

O Teorema 3 pode ser usado como base para um algoritmo para calcular a matriz da relação R^*. Para encontrar essa matriz, são calculadas as potências booleanas sucessivas de $\mathbf{M}_R$, até a potência enésima. Conforme cada potência é calculada, é formada sua conjunção com a conjunção de todas as potências menores. Quando isso é feito com a enésima potência, a matriz para R^* terá sido encontrada. Esse procedimento é mostrado como Algoritmo 1.

ALGORITMO 1 Um Procedimento para Calcular o Fecho Transitivo.

procedure *fecho transitivo* ($\mathbf{M}_R$: matriz zero-um $n \times n$)
$\mathbf{A} := \mathbf{M}_R$
$\mathbf{B} := \mathbf{A}$
for $i := 2$ **to** n
begin
  $\mathbf{A} := \mathbf{A} \odot \mathbf{M}_R$
  $\mathbf{B} := \mathbf{B} \vee \mathbf{A}$
end {$\mathbf{B}$ é a matriz zero-um para R^*)

Podemos facilmente encontrar o número de operações binárias usadas pelo Algoritmo 1 para determinar o fecho transitivo de uma relação. Calcular as potências booleanas $\mathbf{M}_R$, $\mathbf{M}_R^{[2]}$, ..., $\mathbf{M}_R^{[n]}$ exige que sejam feitos $n - 1$ produtos booleanos de matrizes zero-um $n \times n$. Cada um desses produtos booleanos pode ser encontrado usando n^2 $(2n - 1)$ operações binárias. Logo, esses produtos podem ser calculados usando n^2 $(2n - 1)(n - 1)$ operações binárias.

Para encontrar $\mathbf{M}_{R^*}$ a partir de n potências booleanas de $\mathbf{M}_R$, é necessário encontrar $n - 1$ junções de matrizes zero-um. O cálculo de cada uma dessas junções usa n^2 operações binárias. Logo, são usadas $(n - 1)n^2$ operações binárias nesta parte do cálculo. Portanto, quando o Algoritmo 1 é usado, a matriz do fecho transitivo de uma relação em um conjunto com n elementos pode ser encontrada usando $n^2(2n - 1)(n - 1) + (n - 1)n^2 = 2n^3(n - 1)$, que é $O(n^4)$ operações binárias. O restante desta seção descreve um algoritmo mais eficiente para encontrar fechos transitivos.

Algoritmo de Warshall

Links

O algoritmo de Warshall, cujo nome é uma homenagem a Stephen Warshall, que descreveu este algoritmo em 1960, é um método eficiente para calcular o fecho transitivo de uma relação. O Algoritmo 1 pode encontrar o fecho transitivo de uma relação em um conjunto com n elementos usando $2n^3(n - 1)$ operações binárias. Entretanto, o fecho transitivo pode ser encontrado pelo algoritmo de Warshall usando apenas $2n^3$ operações binárias.

Lembre-se: O algoritmo de Warshall algumas vezes é chamado de algoritmo de Roy—Warshall, pois Bernard Roy descreveu este algoritmo em 1959.

Suponha que R seja uma relação em um conjunto com n elementos. Seja $v_1, v_2, \ldots, v_n$ uma listagem arbitrária desses n elementos. O conceito de **vértice interior** de um caminho é usado no algoritmo de Warshall. Se $a, x_1, x_2, \ldots, x_{m-1}, b$ for um caminho, seus vértices interiores são $x_1, x_2, \ldots, x_{m-1}$, ou seja, todos os vértices do caminho que ocorram em algum lugar que não seja nem o primeiro nem o último vértice do caminho. Por exemplo, os vértices interiores de um caminho a, c, d, f, g, h, b, j em um grafo orientado são c, d, f, g, h e b. Os vértices interiores de a, c, d, a, f, b são c, d, a e f. (Observe que o primeiro vértice do caminho não é um vértice interior a menos que ele seja visitado novamente pelo caminho, exceto como último vértice. Analogamente, o último vértice do caminho não é um vértice interior a menos que tenha sido visitado anteriormente pelo caminho, exceto como primeiro vértice.)

O algoritmo de Warshall baseia-se na construção de uma seqüência de matrizes zero-um. Essas matrizes são $\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_n$, em que $\mathbf{W}_0 = \mathbf{M}_R$ é a matriz zero-um dessa relação e $\mathbf{W}_k = \left[w_{ij}^{(k)}\right]$, em que $w_{ij}^{(k)} = 1$ se existir um caminho de v_i a v_j, tal que todos os vértices interiores desse caminho estão no conjunto $\{v_1, v_2, \ldots, v_k\}$ (os primeiros k vértices na lista) e é zero, caso contrário. (O primeiro e o último vértices deste caminho podem estar fora do conjunto dos primeiros k vértices da lista.) Observe que $\mathbf{W}_n = \mathbf{M}_{R^*}$, já que o elemento (i, j) de $\mathbf{M}_{R^*}$ é 1 se e somente se existir um caminho de v_i para v_j, com todos os vértices interiores no conjunto $\{v_1, v_2, \ldots, v_n\}$ (mas esses são os únicos vértices no grafo orientado). O Exemplo 8 ilustra o que a matriz $\mathbf{W}_k$ representa.

EXEMPLO 8 Seja R a relação com grafo orientado mostrado na Figura 3. Sejam a, b, c e d uma listagem dos elementos do conjunto. Encontre as matrizes $\mathbf{W}_0$, $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$ e $\mathbf{W}_4$. A matriz $\mathbf{W}_4$ é o fecho transitivo de R.

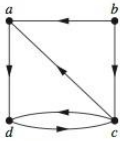


FIGURA 3 O Grafo Orientado da Relação *R*.

Solução: Sejam $v_1 = a$, $v_2 = b$, $v_3 = c$ e $v_4 = d$. $\mathbf{W}_0$ é a matriz da relação. Portanto,

$$\mathbf{W}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

$\mathbf{W}_1$ tem 1 como seu elemento (i, j) se existir um caminho de v_i para v_j que tenha apenas $v_1 = a$ como vértice interior. Observe que todos os caminhos de comprimento um ainda podem ser usados porque eles não têm vértices interiores.

STEPHEN WARSHALL (NASCIDO EM 1935) Stephen Warshall, nascido na cidade de Nova York, freqüentou a escola pública no Brooklyn. Ele estudou na Universidade de Harvard e se formou em matemática em 1956. Nunca recebeu um diploma de pós-graduação, pois naquela época não existia nenhum programa disponível em sua área de interesse. Entretanto, ele fez diversas disciplinas de pós-graduação em diversas universidades e contribuiu para o desenvolvimento da ciência da computação e da engenharia de software.

Depois de se formar em Harvard, Warshall trabalhou no ORO (*Operation Research Office*), que foi montado por Johns Hopkins para fazer pesquisa e desenvolvimento para o Exército norte-americano. Em 1958, ele deixou a ORO para ocupar um cargo em uma companhia chamada Technical Operations, onde ele ajudou a construir um laboratório de pesquisa e desenvolvimento para projetos de software militares. Em 1961, ele deixou a Technical Operations para fundar a Massachusetts Computer Associates. Mais tarde, essa companhia se tornou parte da Applied Data Research (ADR). Depois da fusão, Warshall fez parte do conselho diretor da ADR e gerenciou diversos projetos e organizações. Ele se aposentou da ADR em 1982.

Durante sua carreira, Warshall fez pesquisa e desenvolvimento em sistemas operacionais, projetos de compiladores, projetos de linguagem e pesquisa operacional. No ano acadêmico de 1971-1972, ele apresentou palestras sobre engenharia de software em universidades francesas. Existe uma anedota interessante sobre sua demonstração de que o algoritmo do fecho transitivo, conhecido agora como algoritmo de Warshall, está correto. Ele e um colega da Technical Operations apostaram uma garrafa de rum em quem poderia determinar primeiro se esse algoritmo funcionava sempre. Warshall descobriu sua demonstração durante a noite, ganhando a aposta e o rum, que ele compartilhou com o perdedor da aposta. Como Warshall não gostava de ficar sentado em uma escrivaninha, ele fez a maior parte de seu trabalho criativo em lugares não convencionais, tal como em um barco a vela, no oceano Índico, ou em um pomar de limões grego.

Além disso, existe agora um caminho permitido de b para d, ou seja, b, a, d. Portanto,

$$\mathbf{W}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

$\mathbf{W}_2$ tem 1 como seu elemento (i, j) se existir um caminho de v_i para v_j que tenha apenas $v_1 = a$ e/ou $v_2 = b$ como seus vértices interiores, se houver algum. Como não existem arestas que tenham b como vértice final, nenhum caminho novo é obtido quando permitimos que b seja um vértice interior. Portanto, $\mathbf{W}_2 = \mathbf{W}_1$.

$\mathbf{W}_3$ tem 1 como seu elemento (i, j) se existir um caminho de v_i para v_j que tenha apenas $v_1 = a$, $v_2 = b$ e/ou $v_3 = c$ como seus vértices interiores, se houver algum. Agora temos caminhos de d para a, ou seja, $d, c, a,$ e de d para d, ou seja, d, c, d. Portanto,

$$\mathbf{W}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

Finalmente, $\mathbf{W}_4$ tem 1 como seu elemento (i, j) se existir um caminho de v_i para v_j que tenha apenas $v_1 = a$, $v_2 = b$, $v_3 = c$ e/ou $v_4 = d$ como seus vértices interiores, se houver algum. Como estes são todos os vértices do grafo, este elemento será 1 se e somente se existir um caminho de v_i para v_j. Portanto,

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

Esta última matriz, $\mathbf{W}_4$, é a matriz do fecho transitivo. ◀

O algoritmo de Warshall determina $\mathbf{M}_{R^*}$ calculando eficientemente $\mathbf{W}_0 = \mathbf{M}_R$, $\mathbf{W}_1$, $\mathbf{W}_2$, …, $\mathbf{W}_n = \mathbf{M}_{R^*}$. Esta observação mostra que podemos calcular $\mathbf{W}_k$ diretamente de $\mathbf{W}_{k-1}$: existe um caminho de v_i para v_j sem nenhum vértice além de $v_1, v_2, \ldots, v_k$ como vértices interiores se e somente se ou existe um caminho de v_i para v_j com seus vértices interiores entre os primeiros $k - 1$ vértices na lista ou existem caminhos de v_i para v_k e de v_k para v_j que têm vértices interiores apenas entre os primeiros $k - 1$ vértices da lista. Isto é, ou o caminho de v_i para v_j já existia antes de v_k ser permitido como vértice interior ou permitir v_k como um vértice interior produz um caminho que vai de v_i para v_k e, então, de v_k para v_j. Estes dois casos são mostrados na Figura 4.

O primeiro tipo de caminho existe se e somente se $w_{ij}^{[k-1]} = 1$, e o segundo tipo de caminho existe se e somente se ambos $w_{ik}^{[k-1]}$ e $w_{kj}^{[k-1]}$ são 1. Portanto, $w_{ij}^{[k]}$ é 1 se e somente se ou $w_{ij}^{[k-1]}$ é 1 ou ambos $w_{ik}^{[k-1]}$ e $w_{kj}^{[k-1]}$ são 1. Isto nos dá o Lema 2.

LEMA 2

Seja $\mathbf{W}_k = [w_{ij}^{[k]}]$ a matriz zero-um que tem 1 em sua posição (i, j) se e somente se existir um caminho de v_i para v_j com vértices interiores no conjunto $\{v_1, v_2, \ldots, v_k\}$. Então

$$w_{ij}^{[k]} = w_{ij}^{[k-1]} \vee (w_{ik}^{[k-1]} \wedge w_{kj}^{[k-1]}),$$

sempre que i, j e k forem inteiros positivos que não excedam n.

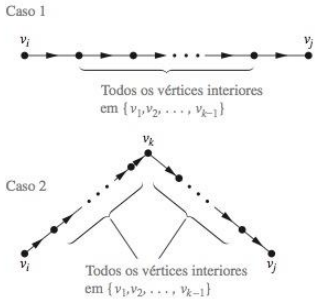


FIGURA 4 A adição de v_k ao Conjunto de Vértices Interiores Permitidos.

O Lema 2 nos dá meios de calcular eficientemente as matrizes $\mathbf{W}_k$, $k = 1, 2, \ldots, n$. Mostramos o pseudocódigo para o algoritmo de Warshall, usando o Lema 2, como Algoritmo 2.

ALGORITMO 2 Algoritmo de Warshall.

```
procedure Warshall (M_R : matriz zero-um n × n)
W := M_R
for k := 1 to n
begin
    for i := 1 to n
    begin
        for j := 1 to n
        w_ij := w_ij ∨ (w_ik ∧ w_kj)
    end
end {W = [w_ij] é M_R*}
```

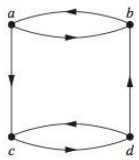
A complexidade computacional do algoritmo de Warshall pode ser facilmente calculada em termos de operações binárias. Para encontrar o elemento $w_{ij}^{[k]}$ a partir dos elementos $w_{ij}^{[k-1]}$, $w_{ik}^{[k-1]}$ e $w_{kj}^{[k-1]}$ usando o Lema 2, são necessárias duas operações binárias. Para encontrar todos os n^2 elementos de $\mathbf{W}_k$ a partir dos elementos de $\mathbf{W}_{k-1}$, são necessárias $2n^2$ operações binárias. Como o algoritmo de Warshall começa com $\mathbf{W}_0 = \mathbf{M}_R$ e calcula a seqüência de n matrizes zero-um $\mathbf{W}_1$, $\mathbf{W}_2$, …, $\mathbf{W}_n = \mathbf{M}_{R^*}$, o número total de operações binárias usadas será $n \cdot 2n^2 = 2n^3$.

Exercícios

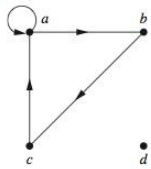
1. Seja R a relação no conjunto $\{0, 1, 2, 3\}$ que contém os pares ordenados $(0, 1)$, $(1, 1)$, $(1, 2)$, $(2, 0)$, $(2, 2)$ e $(3, 0)$. Encontre o
   a) fecho reflexivo de R. **b)** fecho simétrico de R.
2. Seja R a relação $\{(a, b) \mid a \neq b\}$ no conjunto dos inteiros. Qual é o fecho reflexivo de R?
3. Seja R a relação $\{(a, b) \mid a \text{ divide } b\}$ no conjunto dos inteiros. Qual é o fecho simétrico de R?
4. Como pode o grafo orientado que representa o fecho reflexivo de uma relação em um conjunto finito ser construído a partir do grafo orientado da relação?

Nos exercícios 5 a 7, trace o grafo orientado do fecho reflexivo das relações com os grafos orientados mostrados.

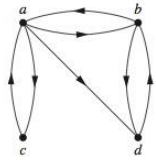
5.



6.



7.



8. Como pode o grafo orientado que representa o fecho simétrico de uma relação em um conjunto finito ser construído a partir do grafo orientado dessa relação?

9. Encontre os grafos orientados dos fechos simétricos das relações com os grafos orientados mostrados nos exercícios 5 a 7.

10. Encontre a menor relação que contém a relação do Exemplo 2 que seja tanto reflexiva quanto simétrica.

11. Encontre o grafo orientado da menor relação que seja tanto reflexiva quanto simétrica para cada uma das relações com grafos orientados mostrados nos exercícios 5 a 7.

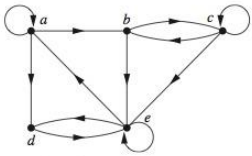
12. Suponha que a relação R no conjunto finito A seja representada pela matriz $\mathbf{M}_R$. Mostre que a matriz que representa o fecho reflexivo de R é $\mathbf{M}_R \vee \mathbf{I}_n$.

13. Suponha que a relação R no conjunto finito A seja representada pela matriz $\mathbf{M}_R$. Mostre que a matriz que representa o fecho simétrico de R é $\mathbf{M}_R \vee \mathbf{M}_R^t$.

14. Mostre que o fecho da relação R relativo à propriedade **P**, se existir, é a interseção de todas as relações com a propriedade **P** que contenham R.

15. Quando é possível definir o "fecho irreflexivo" de uma relação R, isto é, uma relação que contenha R, seja irreflexiva e esteja contida em toda relação irreflexiva que contenha R?

16. Determine se estas seqüências de vértices são caminhos neste grafo orientado.



a) a, b, c, e
b) b, e, c, b, e
c) a, a, b, e, d, e
d) b, c, e, d, a, a, b
e) b, c, c, b, e, d, e, d
f) $a, a, b, b, c, c, b, e, d$

17. Encontre todos os circuitos de comprimento três no grafo orientado do Exercício 16.

18. Determine se existe um caminho no grafo orientado do Exercício 16 que começa no primeiro vértice dado e termina no segundo vértice dado.

a) a, b **b)** b, a **c)** b, b
d) a, e **e)** b, d **f)** c, d
g) d, d **h)** e, a **i)** e, c

19. Seja R a relação no conjunto $\{1, 2, 3, 4, 5\}$ que contém os pares ordenados $(1, 3)$, $(2, 4)$, $(3, 1)$, $(3, 5)$, $(4, 3)$, $(5, 1)$, $(5, 2)$ e $(5, 4)$. Encontre

a) R^2. **b)** R^3. **c)** R^4.
d) R^5. **e)** R^6. **f)** R^*.

20. Seja R a relação que contém o par (a, b) se a e b forem cidades, tal que existe um vôo direto sem escalas de a para b. Quando (a, b) está em

a) R^2? **b)** R^3? **c)** R^*?

21. Seja R a relação no conjunto de todos os estudantes que contém o par ordenado (a, b) se a e b tiverem, pelo menos, uma aula em comum e $a \neq b$. Quando (a, b) está em

a) R^2? **b)** R^3? **c)** R^*?

22. Suponha que a relação R seja reflexiva. Mostre que R^* é reflexiva.

23. Suponha que a relação R seja simétrica. Mostre que R^* é simétrica.

24. Suponha que a relação R seja irreflexiva. A relação R^2 é necessariamente irreflexiva?

25. Use o Algoritmo 1 para encontrar os fechos transitivos destas relações em $\{1, 2, 3, 4\}$.

a) $\{(1, 2) (2, 1), (2, 3), (3, 4), (4, 1)\}$
b) $\{(2, 1), (2, 3), (3, 1), (3, 4), (4, 1), (4, 3)\}$
c) $\{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$
d) $\{(1, 1), (1, 4), (2, 1), (2, 3), (3, 1), (3, 2), (3, 4), (4, 2)\}$

26. Use o Algoritmo 1 para encontrar os fechos transitivos destas relações em $\{a, b, c, d, e\}$.

a) $\{(a, c), (b, d), (c, a), (d, b), (e, d)\}$
b) $\{(b, c), (b, e), (c, e), (d, a), (e, b), (e, c)\}$
c) $\{(a, b), (a, c), (a, e), (b, a), (b, c), (c, a), (c, b), (d, a), (e, d)\}$
d) $\{(a, e), (b, a), (b, d), (c, d), (d, a), (d, c), (e, a), (e, b), (e, c), (e, e)\}$

27. Use o algoritmo de Warshall para encontrar os fechos transitivos das relações do Exercício 25.

28. Use o algoritmo de Warshall para encontrar os fechos transitivos das relações do Exercício 26.

29. Encontre a menor relação que contém a relação $\{(1, 2), (1, 4), (3, 3), (4, 1)\}$ que seja

a) reflexiva e transitiva.

b) simétrica e transitiva.
c) reflexiva, simétrica e transitiva.

30. Termine a demonstração do caso em que $a \neq b$ no Lema 1.

31. Foram desenvolvidos algoritmos que usam $O(n^{2,8})$ operações binárias para calcular o produto booleano de duas matrizes zero-um $n \times n$. Supondo que esses algoritmos possam ser usados, dê estimativas O grande do número de operações binárias usando o Algoritmo 1 e usando o algoritmo de Warshall para encontrar o fecho transitivo de uma relação em um conjunto com n elementos.

***32.** Desenvolva um algoritmo usando o conceito de vértices interiores em um caminho para encontrar o caminho mais curto entre dois vértices em um grafo orientado, se esse caminho existir.

33. Adapte o Algoritmo 1 para encontrar o fecho reflexivo do fecho transitivo de uma relação em um conjunto com n elementos.

34. Adapte o algoritmo de Warshall para encontrar o fecho reflexivo do fecho transitivo de uma relação em um conjunto com n elementos.

35. Mostre que o fecho relativo à propriedade **P** de uma relação $R = \{(0, 0), (0, 1), (1, 1), (2, 2)\}$ no conjunto $\{0, 1, 2\}$ não existe se **P** for a propriedade que
a) "não é reflexiva".
b) "tem um número ímpar de elementos".

8.5 Relações de Equivalência

Introdução

Em algumas linguagens de programação, o nome das variáveis pode conter um número ilimitado de caracteres. Entretanto, há um limite no número de caracteres que são verificados quando um compilador determina se duas variáveis são iguais. Por exemplo, no C tradicional, apenas os primeiros oito caracteres do nome de uma variável são verificados pelo compilador. (Esses caracteres são letras maiúsculas ou minúsculas, algarismos ou sublinhas.) Conseqüentemente, o compilador considera que seqüências de mais de oito caracteres que coincidam em seus oito primeiros caracteres são a mesma coisa. Seja R a relação no conjunto das seqüências de caracteres, tal que $s\ R\ t$, em que s e t são duas seqüências, se s e t tiverem, pelo menos, oito caracteres de comprimento e os primeiros oito caracteres coincidirem ou $s = t$. É fácil perceber que R é reflexiva, simétrica e transitiva. Além disso, R divide o conjunto das seqüências em classes, de modo que todas as seqüências em uma classe particular são consideradas a mesma coisa por um compilador para o C tradicional.

Os inteiros a e b estão relacionados pela relação de "congruência módulo 4" quando 4 dividir $a - b$. Mostraremos mais adiante que essa relação é reflexiva, simétrica e transitiva. Não é difícil perceber que a está relacionado a b se e somente se a e b tiverem o mesmo resto quando divididos por 4. Segue que esta é uma relação que separa o conjunto dos inteiros em quatro classes diferentes. Quando estivermos interessados apenas em que resto um inteiro deixa quando dividido por 4, precisamos apenas saber a qual classe ele pertence, não seu valor particular.

Estas duas relações, R e a congruência módulo 4, são exemplos de relações de equivalência, ou seja, relações que são reflexivas, simétricas e transitivas. Nesta seção mostraremos que essas relações dividem conjuntos em classes disjuntas de elementos equivalentes. As relações de equivalência aparecem sempre que estivermos interessados apenas em se um elemento de um conjunto está em determinada classe de elementos, em vez de nos preocuparmos sobre sua identidade particular.

Relações de Equivalência

Nesta seção estudaremos relações com uma combinação particular de propriedades que permite que elas sejam usadas para relacionar objetos que são parecidos de alguma maneira.

DEFINIÇÃO 1 Uma relação em um conjunto A é denominada uma *relação de equivalência* se for reflexiva, simétrica e transitiva.

Relações de equivalência são importantes em toda a matemática e na ciência da computação. Uma razão para isso é que em uma relação de equivalência, quando dois elementos estão relacionados, faz sentido dizer que são equivalentes.

DEFINIÇÃO 2 Dois elementos a e b que estão relacionados por uma relação de equivalência são denominados *equivalentes*. A notação $a \sim b$ é freqüentemente usada para indicar que a e b são elementos equivalentes com referência a uma relação de equivalência particular.

Para que a noção de elementos equivalentes faça sentido, todo elemento deve ser equivalente a si próprio, como garante a propriedade reflexiva para uma relação de equivalência. Faz sentido dizer que a e b estão relacionados (e não apenas que a está relacionado a b) para uma relação de equivalência, pois quando a estiver relacionado a b, pela propriedade de simetria, b estará relacionado a a. Além disso, como uma relação de equivalência é transitiva, se a e b forem equivalentes e b e c também forem, segue que a e c serão equivalentes.

Os exemplos 1 a 5 ilustram a noção de relação de equivalência.

EXEMPLO 1 Seja R a relação no conjunto dos inteiros, tal que $a\ R\ b$ se e somente se $a = b$ ou $a = -b$. Na seção 8.1 mostramos que R é reflexiva, simétrica e transitiva. Segue que R é uma relação de equivalência. ◀

EXEMPLO 2 Seja R a relação no conjunto dos números reais, tal que $a\ R\ b$ se e somente se $a - b$ for um inteiro. R é uma relação de equivalência?

Exemplos Extras

Solução: Como $a - a = 0$ é um inteiro para todos os números reais a, $a\ R\ a$ para todos os números a. Portanto, R é reflexiva. Suponha agora que $a\ R\ b$. Então, $a - b$ é um inteiro, de modo que $b - a$ também é um inteiro. Logo, $b\ R\ a$. Segue que R é simétrica. Se $a\ R\ b$ e $b\ R\ c$, então $a - b$ e $b - c$ são inteiros. Portanto, $a - c = (a - b) + (b - c)$ também é um inteiro. Assim, $a\ R\ c$. Logo, R é transitiva. Conseqüentemente, R é uma relação de equivalência. ◀

Uma das relações de equivalência mais amplamente usadas é a congruência módulo m, em que m é um inteiro positivo maior que 1.

EXEMPLO 3 **Congruência Módulo m** Seja m um inteiro positivo com $m > 1$. Mostre que a relação

$$R = \{(a, b) \mid a \equiv b \pmod{m}\}$$

é uma relação de equivalência no conjunto dos inteiros.

Solução: Lembre da Seção 3.4, em que $a \equiv b \pmod{m}$ se e somente se m divide $a - b$. Observe que $a - a = 0$ é divisível por m, pois $0 = 0 \cdot$ m. Portanto, $a \equiv a \pmod{m}$, de modo que a congruência módulo m é reflexiva. Suponha agora que $a \equiv b \pmod{m}$. Então, $a - b$ é divisível por m, de modo que $a - b = km$, em que k é um inteiro. Segue que $b - a = (-k)m$, de modo que $b \equiv a \pmod{m}$. Portanto, a congruência módulo m é simétrica. A seguir, suponha que $a \equiv b \pmod{m}$ e $b \equiv c \pmod{m}$. Então, m divide tanto $a - b$ quanto $b - c$. Portanto, existem inteiros k e l com $a - b = km$ e $b - c = lm$. Somando estas duas equações, obtém-se $a - c = (a - b) + (b - c) = km + lm = (k + l)m$. Logo, $a \equiv c \pmod{m}$. Portanto, a congruência módulo m é transitiva. Segue que a congruência módulo m é uma relação de equivalência. ◀

EXEMPLO 4 Suponha que R seja a relação no conjunto das seqüências de letras do inglês tal que $a\ R\ b$ se e somente se $l(a) = l(b)$, em que $l(x)$ é o comprimento da seqüência x. R é uma relação de equivalência?

Solução: Como $l(a) = l(a)$, segue que $a\ R\ a$ sempre que a for uma seqüência, de modo que R é reflexiva. A seguir, suponha que $a\ R\ b$, de modo que $l(a) = l(b)$. Então, $b\ R\ a$, pois $l(b) = l(a)$. Assim, R é simétrica. Finalmente, suponha que $a\ R\ b$ e $b\ R\ c$. Então, $l(a) = l(b)$ e $l(b) = l(c)$. Portanto, $l(a) = l(c)$, de modo que $a\ R\ c$. Conseqüentemente, R é transitiva. Como R é reflexiva, simétrica e transitiva, ela é uma relação de equivalência. ◀

EXEMPLO 5 Seja n um inteiro positivo e S um conjunto de seqüências. Suponha que R_n seja a relação em S tal que $s\ R_n t$ se e somente se $s = t$ ou tanto s quanto t têm, pelo menos, n caracteres e os primeiros n caracteres de s e de t são os mesmos. Isto é, uma seqüência de menos de n caracteres está relacionada apenas a ela mesma; uma seqüência s com, pelo menos, n caracteres está relacionada a uma seqüência t se e somente se t tiver, pelo menos, n caracteres e t começar com os n caracteres no início de s. Por exemplo, seja n = 3 e seja S o conjunto de todas as seqüências de bits. Então, $s\ R_3 t$ ou quando $s = t$ ou quando s e t forem seqüências de bits de comprimento maior que ou igual a três que comecem com os mesmos primeiros 3 bits. Por exemplo, 01 R_3 01 e 00111 R_3 00101, mas 01 $\not R_3$ 010 e 01011 $\not R_3$ 01110.

Mostre que para todo conjunto S de seqüências e todo inteiro positivo n, R_n é um a relação de equivalência em S.

Solução: A relação R_n é reflexiva porque $s = s$, de modo que $s\ R_n s$ sempre que s for uma seqüência em S. Se $s\ R_n t$, então ou $s = t$ ou ambos s e t têm, pelo menos, n caracteres de comprimento e começam com os mesmos n caracteres. Isso significa que $t\ R_n s$. Concluímos que R_n é simétrica.

Suponha agora que $s\ R_n t$ e $t\ R_n u$. Então, $s = t$ ou s e t têm, pelo menos, n caracteres de comprimento e s e t começam com os mesmos n caracteres, e ou $t = u$ ou t e u têm, pelo menos, n caracteres de comprimento e t e u começam com os mesmos n caracteres. A partir disso, podemos deduzir que ou $s = u$ ou ambos, s e u, têm n caracteres de comprimento e s e u começam com os mesmos n caracteres (porque, nesse caso, sabemos que s, t e u têm, pelo menos, n caracteres de comprimento e ambos, s e u, começam com os mesmos n caracteres que t começa). Conseqüentemente, R_n é transitiva. Segue que R_n é uma relação de equivalência. ◀

Nos exemplos 6 e 7, veremos duas relações que não são relações de equivalência.

EXEMPLO 6 Mostre que a relação "divide" no conjunto dos inteiros positivos não é uma relação de equivalência.

Solução: Pelos exemplos 9 e 15 da Seção 8.1, sabemos que a relação "divide" é reflexiva e transitiva. Entretanto, pelo Exemplo 12 da Seção 8.1, sabemos que essa relação não é simétrica (por exemplo, $2 \mid 4$, mas $4 \nmid 2$). Concluímos que a relação "divide" no conjunto dos inteiros positivos não é uma relação de equivalência. ◀

EXEMPLO 7 Seja R a relação no conjunto dos números reais tal que $x\ R\ y$ se e somente se x e y forem números reais que difiram por menos que 1, isto é, $|x - y| < 1$. Mostre que R não é uma relação de equivalência.

Solução: R é reflexiva, pois $|x - x| = 0 < 1$ sempre que $x \in \mathbf{R}$. R é simétrica, pois se $x\ R\ y$, em que x e y são números reais, então $|x - y| < 1$, o que nos diz que $|y - x| = |x - y| < 1$, de modo que $y R x$. Entretanto, R não é uma relação de equivalência porque não é transitiva. Considere $x = 2{,}8$, $y = 1{,}9$ e $z = 1{,}1$, de modo que $|x - y| = |2{,}8 - 1{,}9| = 0{,}9 < 1$, $|y - z| = |1{,}9 - 1{,}1| = 0{,}8 < 1$, mas $|x - z| = |2{,}8 - 1{,}1| = 1{,}7 > 1$. Ou seja, $2{,}8\ R\ 1{,}9$ e $1{,}9\ R\ 1{,}1$, mas $2{,}8 \not R\ 1{,}1$. ◀

Classes de Equivalência

Seja A o conjunto de todos os estudantes de sua escola que se formaram no ensino médio. Considere a relação R em A que consiste em todos os pares (x, y), em que x e y se formaram na mesma escola no ensino médio. Dado um estudante x, podemos formar o conjunto de todos os estudantes equivalentes a x relativo a R. Esse conjunto consiste em todos os estudantes que se formaram na mesma escola de ensino médio que x. Esse subconjunto de A é chamado de classe de equivalência da relação.

DEFINIÇÃO 3 Seja R uma relação de equivalência em um conjunto A. O conjunto de todos os elementos que estão relacionados a um elemento a de A é chamado de *classe de equivalência* de a. A classe de equivalência de a relativa a R é indicada por $[a]_R$. Quando apenas uma relação está sendo considerada, podemos omitir o R subscrito e escrever $[a]$ para essa classe de equivalência.

Em outras palavras, se R for uma relação de equivalência em um conjunto A, a classe de equivalência do elemento a é

$$[a]_R = \{s \mid (a, s) \in R\}.$$

Se $b \in [a]_R$, então b é denominado um **representante** dessa classe de equivalência. Qualquer elemento de uma classe pode ser usado como um representante dessa classe. Ou seja, não há nada especial sobre o elemento particular escolhido como o representante da classe.

EXEMPLO 8 Qual é a classe de equivalência de um inteiro para a relação de equivalência do Exemplo 1?

Solução: Como um inteiro é equivalente a ele mesmo e a seu oposto nessa relação de equivalência, segue que $[a] = \{-a, a\}$. Esse conjunto contém dois inteiros distintos, a menos que $a = 0$. Por exemplo, $[7] = \{-7, 7\}$, $[-5] = \{-5, 5\}$ e $[0] = \{0\}$. ◀

EXEMPLO 9 Quais são as classes de equivalência de 0 e 1 na congruência módulo 4?

Solução: A classe de equivalência de 0 contém todos os inteiros a, tal que $a \equiv 0 \pmod 4$. Os inteiros nessa classe são aqueles divisíveis por 4. Portanto, a classe de equivalência de 0 nesta relação é

$$[0] = \{\ldots, -8, -4, 0, 4, 8, \ldots\}.$$

A classe de equivalência de 1 contém todos os inteiros a, tal que $a \equiv 1 \pmod 4$. Os inteiros nessa classe são aqueles que, quando divididos por 4, têm resto 1. Portanto, a classe de equivalência de 1 nesta relação é

$$[1] = \{\ldots, -7, -3, 1, 5, 9, \ldots\}.$$ ◀

No Exemplo 9, as classes de equivalência de 0 e 1 relativas à congruência módulo 4 foram encontradas. O Exemplo 9 pode ser facilmente generalizado, substituindo-se 4 por qualquer inteiro positivo m. As classes de equivalência da relação de congruência módulo m são chamadas de **classes de congruência módulo** m. A classe de congruência de um inteiro a módulo m é indicada por $[a]_m$, de modo que $[a]_m = \{\ldots, a - 2m, a - m, a, a + m, a + 2m, \ldots\}$. Como ilustração, segue do Exemplo 9 que $[0]_4 = \{\ldots, -8, -4, 0, 4, 8, \ldots\}$ e $[1]_4 = \{\ldots, -7, -3, 1, 5, 9, \ldots\}$.

EXEMPLO 10 Qual é a classe de equivalência da seqüência 0111 relativa à relação de equivalência R_3 do Exemplo 5, no conjunto de todas as seqüências de bits? (Lembre-se que sR_3t se e somente se s e t forem seqüências de bits com $s = t$ ou s e t forem seqüências com, pelo menos, três bits que comecem com os mesmos três bits.)

Solução: As seqüências equivalentes a 0111 são as seqüências de bits com, pelo menos, três bits que comecem com 011. Estas são as seqüências de bits 011, 0110, 0111, 01100, 01101, 01110, 01111, e assim por diante. Conseqüentemente,

$$[011]_{R_3} = \{011, 0110, 0111, 01100, 01101, 01110, 01111 \ldots\}.$$ ◀

EXEMPLO 11 **Identificadores na Linguagem de Programação C** Na linguagem de programação C, um **identificador** é o nome de uma variável, de uma função ou de outro tipo de entidade. Cada identificador é um conjunto não vazio de caracteres no qual cada caractere é uma letra minúscula ou maiúscula da língua inglesa, um algarismo ou uma sublinha, e o primeiro caractere é uma letra maiúscula ou minúscula da língua inglesa. Os identificadores podem ser de qualquer comprimento. Isso permite que os programadores usem tantos caracteres quantos quiserem para dar nome a uma entidade, como uma variável. Entretanto, para os compiladores para algumas versões de *C*, existe um limite no número de caracteres verificado quando dois nomes são comparados para ver se eles se referem à mesma coisa. Por exemplo, os compiladores de Standard C consideram dois identificadores como o mesmo, quando os seus primeiros 31 caracteres coincidirem. Conseqüentemente, os programadores devem ter cuidado para não usar os mesmos 31 caracteres iniciais para coisas diferentes. Vemos que dois identificadores são considerados o mesmo quando estão relacionados pela relação R_{31} do Exemplo 5. Usando o Exemplo 5, sabemos que R_{31}, no conjunto de todos os identificadores no Standard C, é uma relação de equivalência.

Quais são as classes de equivalência de cada um dos identificadores Number_of_tropical_storms, Number_of_named_tropical_storms e Number_of_named_tropical_storms_in_the_Atlantic_in_2005?

Solução: Observe que quando um identificador tiver menos do que 31 caracteres, pela definição de R_{31}, sua classe de equivalência contém apenas ele mesmo. Como o identificador Number_of_tropical_storms tem 25 caracteres de comprimento, sua classe de equivalência conterá apenas ele mesmo.

O identificador Number_of_named_tropical_storms tem exatamente 31 caracteres de comprimento. Um identificador é equivalente a ele quando começa com os mesmos 31 caracteres. Conseqüentemente, todo identificador de, pelo menos, 31 caracteres de comprimento que comece com Number_of_named_tropical_storms é equivalente a esse identificador. Segue que a classe de equivalência de Number_of_named_tropical_storms é o conjunto de todos os identificadores que começam com os 31 caracteres Number_of_named_tropical_storms.

Um identificador é equivalente a Number_of_named_tropical_storms_in_the_Atlantic_in_2005 se e somente se ele começa com seus primeiros 31 caracteres. Como esses caracteres são Number_of_named_tropical_storms, vemos que um identificador será equivalente a Number_of_named_tropical_storms_in_the_Atlantic_in_2005 se e somente se for equivalente a Number_of_named_tropical_storms. Segue que estes dois últimos identificadores têm a mesma classe de equivalência. ◀

Classes de Equivalência e Partições

Seja A o conjunto dos estudantes de sua escola que estão fazendo apenas um curso, e seja R a relação em A que consiste nos pares (x, y), em que x e y são estudantes que fazem o mesmo curso. Então, R é uma relação de equivalência, como o leitor deve verificar. Podemos ver que R divide todos os estudantes em A em uma coleção disjunta de subconjuntos, em que cada subconjunto contém os alunos que fazem um curso especificado. Por exemplo, um subconjunto contém todos os estudantes que fazem ciência da computação (e nenhum outro curso), e um segundo subconjunto contém todos os estudantes de história. Além disso, estes subconjuntos são

classes de equivalência de R. Este exemplo ilustra como as classes de equivalência de uma relação de equivalência dividem um conjunto em subconjuntos disjuntos, não vazios. Tornaremos essas noções mais precisas na discussão a seguir.

Seja R uma relação de equivalência em um conjunto A. O Teorema 1 mostra que as classes de equivalência de dois elementos de A ou são idênticas ou são disjuntas.

TEOREMA 1

Seja R uma relação de equivalência em um conjunto A. Estas afirmações para elementos a e b de A são equivalentes:

(*i*) $a\,R\,b$ (*ii*) $[a] = [b]$ (*iii*) $[a] \cap [b] \neq \emptyset$

***Demonstração*:** Vamos primeiro mostrar que (*i*) implica (*ii*). Suponha que $a\,R\,b$. Vamos demonstrar que $[a] = [b]$ mostrando que $[a] \subseteq [b]$ e $[b] \subseteq [a]$. Suponha que $c \in [a]$. Então, $a\,R\,c$. Como $a\,R\,b$ e R é simétrica, sabemos que $b\,R\,a$. Além disso, como R é transitiva e $b\,R\,a$ e $a\,R\,c$, segue que $b\,R\,c$. Portanto, $c \in [b]$. Isto mostra que $[a] \subseteq [b]$. A demonstração de que $[b] \subseteq [a]$ é análoga; ela será deixada como um exercício para o leitor.

Segundo, vamos mostrar que (*ii*) implica (*iii*). Suponha que $[a] = [b]$. Segue que $[a] \cap [b] \neq \emptyset$, pois $[a]$ é não vazia (pois $a \in [a]$, uma vez que R é reflexiva).

A seguir, vamos mostrar que (*iii*) implica (*i*). Suponha que $[a] \cap [b] \neq \emptyset$. Então, existe um elemento c, com $c \in [a]$ e $c \in [b]$. Em outras palavras, $a\,R\,c$ e $b\,R\,c$. Pela propriedade simétrica, $c\,R\,b$. Então, por transitividade, como $a\,R\,c$ e $c\,R\,b$, temos $a\,R\,b$.

Como (*i*) implica (*ii*), (*ii*) implica (*iii*) e (*iii*) implica (*i*), as três afirmações, (*i*), (*ii*) e (*iii*), são equivalentes. ◁

Nós agora estamos em posição de mostrar como uma relação de equivalência *divide* um conjunto. Seja R uma relação de equivalência em um conjunto A. A união das classes de equivalência de R é todo o A, pois um elemento a de A está em sua própria classe de equivalência, ou seja, $[a]_R$. Em outras palavras,

$$\bigcup_{a \in A} [a]_R = A.$$

Além disso, pelo Teorema 1, segue que estas classes de equivalência são ou iguais ou disjuntas, de modo que

$$[a]_R \cap [b]_R = \emptyset,$$

quando $[a]_R \neq [b]_R$.

Essas duas observações mostram que as classes de equivalência formam uma partição de A, pois separam A em subconjuntos disjuntos. Mais precisamente, uma **partição** de um conjunto S é uma coleção de subconjuntos não vazios disjuntos de S, cuja união é S. Em outras palavras, a coleção de subconjuntos A_i, $i \in I$ (em que I é um conjunto de índices) forma uma partição de S se e somente se

$$A_i \neq \emptyset \text{ para } i \in I,$$

$$A_i \cap A_j = \emptyset \text{ quando } i \neq j$$

e

$$\bigcup_{i \in I} A_i = S.$$

(Aqui, a notação $\cup_{i \in I} A_i$ representa a união dos conjuntos A_i para todo $i \in I$.) A Figura 1 ilustra o conceito da partição de um conjunto.

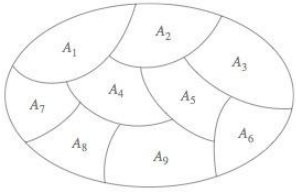


FIGURA 1 **Uma Partição de um Conjunto.**

EXEMPLO 12 Suponha que $S = \{1, 2, 3, 4, 5, 6\}$. A coleção de conjuntos $A_1 = \{1, 2, 3\}$, $A_2 = \{4, 5\}$ e $A_3 = \{6\}$ forma uma partição de S, pois esses conjuntos são disjuntos e sua união é S. ◀

Vimos que as classes de equivalência de uma relação de equivalência em um conjunto formam uma partição do conjunto. Os subconjuntos nessa partição são classes de equivalência. Reciprocamente, toda partição de um conjunto pode usada para formar uma relação de equivalência. Dois elementos são equivalentes relativamente a essa relação se e somente se eles estiverem no mesmo subconjunto da partição.

Para ver isso, suponha que $\{A_i \mid i \in I\}$ seja uma partição de um conjunto S. Seja R a relação em S que consiste nos pares (x, y), em que x e y pertencem ao mesmo subconjunto A_i da partição. Para mostrar que R é uma relação de equivalência, devemos mostrar que R é reflexiva, simétrica e transitiva.

Vemos que $(a, a) \in R$ para todo $a \in S$, pois a está no mesmo subconjunto que ele mesmo. Logo, R é reflexiva. Se $(a, b) \in R$, então b e a estão no mesmo subconjunto da partição, de modo que $(b, a) \in R$ também. Logo, R é simétrica. Se $(a, b) \in R$ e $(b, c) \in R$, então a e b estão em um mesmo subconjunto X da partição e b e c estão em um mesmo subconjunto Y da partição. Como os subconjuntos da partição são disjuntos e b pertence a X e a Y, segue que $X = Y$. Conseqüentemente, a e c pertencem ao mesmo subconjunto da partição, de modo que $(a, c) \in R$. Logo, R é transitiva.

Segue que R é uma relação de equivalência. As classes de equivalência de R consistem nos subconjuntos de S que contêm elementos relacionados e, pela definição de R, esses são os subconjuntos da partição. O Teorema 2 resume as conexões que estabelecemos entre relações de equivalência e partições.

TEOREMA 2 Seja R uma relação de equivalência em um conjunto S. Então, as classes de equivalência de R formam uma partição de S. Reciprocamente, dada uma partição $\{A_i \mid i \in I\}$ do conjunto S, existe uma relação de equivalência R que tem os conjuntos A_i, $i \in I$ como suas classes de equivalência.

O Exemplo 13 mostra como construir uma relação de equivalência a partir de uma partição.

EXEMPLO 13 Liste os pares ordenados na relação de equivalência R produzidos pela partição $A_1 = \{1, 2, 3\}$, $A_2 = \{4, 5\}$ e $A_3 = \{6\}$ de $S = \{1, 2, 3, 4, 5, 6\}$, dada no Exemplo 12.

Solução: Os subconjuntos na partição são as classes de equivalência de R. O par $(a, b) \in R$ se e somente se a e b estiverem no mesmo subconjunto da partição. Os pares (1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2) e (3, 3) pertencem a R, pois $A_1 = \{1, 2, 3\}$ é uma classe de equivalência; os pares (4, 4), (4, 5), (5, 4) e (5, 5) pertencem a R porque $A_2 = \{4, 5\}$ é uma classe de equivalência; e, finalmente, o par (6, 6) pertence a R porque $\{6\}$ é uma classe de equivalência. Nenhum outro par além desses citados pertence a R. ◀

As classes de congruência módulo m fornecem uma ilustração útil do Teorema 2. Existem m classes de congruência módulo m diferentes, correspondentes aos m restos diferentes possíveis

quando um inteiro é dividido por m. Essas m classes de congruência são indicadas por $[0]_m$, $[1]_m$, ..., $[m-1]_m$. Elas formam uma partição do conjunto dos inteiros.

EXEMPLO 14 Quais são os conjuntos na partição dos inteiros que aparecem na congruência módulo 4?

Solução: Existem quatro classes de congruência, que correspondem a $[0]_4$, $[1]_4$, $[2]_4$ e $[3]_4$. Elas são os conjuntos

$$[0]_4 = \{\ldots, -8, -4, 0, 4, 8, \ldots\},$$
$$[1]_4 = \{\ldots, -7, -3, 1, 5, 9, \ldots\},$$
$$[2]_4 = \{\ldots, -6, -2, 2, 6, 10, \ldots\},$$
$$[3]_4 = \{\ldots, -5, -1, 3, 7, 11, \ldots\}.$$

Essas classes de congruência são disjuntas, e todo inteiro está exatamente em uma delas. Em outras palavras, como o Teorema 2 mostra, essas classes de congruência formam uma partição. ◀

Fornecemos agora um exemplo de uma partição do conjunto de todas as sequências resultantes de uma relação de equivalência neste conjunto.

EXEMPLO 15 Seja R_3 a relação de equivalência do Exemplo 5. Quais são os conjuntos na partição do conjunto de todas as seqüências de bits resultantes da relação R_3 no conjunto de todas as seqüências de bits? (Lembre-se de que $s\,R_3\,t$, em que s e t são seqüências de bits, se $s = t$ ou se s e t forem seqüências de bits com, pelo menos, três bits que coincidam nos seus primeiros três bits.)

Solução: Observe que toda seqüência de bits de comprimento menor do que 3 é equivalente apenas a ela própria. Logo, $[\lambda]_{R_3} = \{\lambda\}$, $[0]_{R_3} = \{0\}$, $[1]_{R_3} = \{1\}$, $[00]_{R_3} = \{00\}$, $[01]_{R_3} = \{01\}$, $[10]_{R_3} = \{10\}$ e $[11]_{R_3} = \{11\}$. Observe que toda seqüência de bits de comprimento maior que ou igual a 3 é equivalente a uma das oito seqüências de bits 000, 001, 010, 011, 100, 101, 110 e 111. Temos

$$[000]_{R_3} = \{000, 0000, 0001, 00000, 00001, 00010, 00011, \ldots\},$$
$$[001]_{R_3} = \{001, 0010, 0011, 00100, 00101, 00110, 00111, \ldots\},$$
$$[010]_{R_3} = \{010, 0100, 0101, 01000, 01001, 01010, 01011, \ldots\},$$
$$[011]_{R_3} = \{011, 0110, 0111, 01100, 01101, 01110, 01111, \ldots\},$$
$$[100]_{R_3} = \{100, 1000, 1001, 10000, 10001, 10010, 10011, \ldots\},$$
$$[101]_{R_3} = \{101, 1010, 1011, 10100, 10101, 10110, 10111, \ldots\},$$
$$[110]_{R_3} = \{110, 1100, 1101, 11000, 11001, 11010, 11011, \ldots\}, \text{ e}$$
$$[111]_{R_3} = \{111, 1110, 1111, 11100, 11101, 11110, 11111, \ldots\}.$$

Essas 15 classes de equivalência são disjuntas e toda seqüência de bits está exatamente em uma delas. Como o Teorema 2 nos mostra, essas classes de equivalência formam uma partição do conjunto de todas as seqüências de bits. ◀

Exercícios

1. Quais destas relações em {0, 1, 2, 3} são relações de equivalência? Determine as propriedades de uma relação de equivalência que faltam para as outras.

a) {(0, 0), (1, 1), (2, 2), (3, 3)}
b) {(0, 0), (0, 2), (2, 0), (2, 2), (2, 3), (3, 2), (3, 3)}
c) {(0, 0), (1, 1), (1, 2), (2, 1), (2, 2), (3, 3)}

d) {(0, 0), (1, 1), (1, 3), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)}
e) {(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 2), (3, 3)}

2. Quais destas relações no conjunto de todas as pessoas são relações de equivalência? Determine as propriedades de uma relação de equivalência que faltam para as outras.
a) {(*a*, *b*) | *a* e *b* têm a mesma idade}
b) {(*a*, *b*) | *a* e *b* têm os mesmos pais}
c) {(*a*, *b*) | *a* e *b* têm um progenitor em comum}
d) {(*a*, *b*) | *a* e *b* já se encontraram}
e) {(*a*, *b*) | *a* e *b* falam uma mesma língua}

3. Quais destas relações no conjunto de todas as funções de **Z** para **Z** são relações de equivalência? Determine as propriedades de uma relação de equivalência que faltam para as outras.
a) $\{(f, g) \mid f(1) = g(1)\}$
b) $\{(f, g) \mid f(0) = g(0)$ ou $f(1) = g(1)\}$
c) $\{(f, g) \mid f(x) - g(x) = 1$ para todo $x \in \mathbf{Z}\}$
d) $\{(f, g) \mid$ para algum $C \in \mathbf{Z}$, para todo $x \in \mathbf{Z}$, $f(x) - g(x) = C\}$
e) $\{(f, g) \mid f(0) = g(1)$ e $f(1) = g(0)\}$

4. Defina três relações de equivalência no conjunto dos estudantes de sua aula de matemática discreta, diferentes das relações discutidas no texto. Determine as classes de equivalência para cada uma dessas relações de equivalência.

5. Defina três relações de equivalência no conjunto de prédios de um *campus* universitário. Determine as classes de equivalência para cada uma dessas relações de equivalência.

6. Defina três relações de equivalência no conjunto de disciplinas oferecidas em sua escola. Determine as classes de equivalência de cada uma dessas relações de equivalência.

7. Mostre que a relação de equivalência lógica no conjunto de todas as proposições compostas é uma relação de equivalência. Quais são as classes de equivalência de **V** e **F**?

8. Seja R a relação no conjunto de todos os conjuntos de números reais tal que $S\,R\,T$ se e somente se S e T tiverem a mesma cardinalidade. Mostre que R é uma relação de equivalência. Quais são as classes de equivalência dos conjuntos {0, 1, 2} e **Z**?

9. Suponha que A seja um conjunto não vazio e que f seja uma função que tem A como seu domínio. Seja R a relação em A que consiste em todos os pares ordenados (x, y) tal que $f(x) = f(y)$.
a) Mostre que R é uma relação de equivalência em A.
b) Quais são as classes de equivalência de R?

10. Suponha que A seja um conjunto não vazio e que R seja uma relação de equivalência em A. Mostre que existe uma função f com A como seu domínio tal que $(x, y) \in R$ se e somente se $f(x) = f(y)$.

11. Mostre que a relação R, que consiste em todos os pares (x, y) tal que x e y são seqüências de bits de comprimento maior que ou igual a 3, que coincidem em seus primeiros três bits, é uma relação de equivalência no conjunto de todas as seqüências de bits de comprimento maior que ou igual a 3.

12. Mostre que a relação R, que consiste em todos os pares (x, y) tal que x e y são seqüências de bits de comprimento maior que ou igual a 3, que coincidem, exceto, possivelmente, em seus primeiros três bits, é uma relação de equivalência no conjunto de todas as seqüências de bits de comprimento maior que ou igual a 3.

13. Mostre que a relação R, que consiste em todos os pares (x, y) tal que x e y são seqüências de bits que coincidem nos seus primeiro e terceiro bits, é uma relação de equivalência no conjunto de todas as seqüências de bits de comprimento maior que ou igual a 3.

14. Seja R a relação que consiste em todos os pares (x, y) tal que x e y são seqüências de letras maiúsculas e minúsculas da língua inglesa com a propriedade que, para todo inteiro positivo n, o enésimo caractere em x e y é a mesma letra, ou minúscula ou maiúscula. Mostre que R é uma relação de equivalência.

15. Seja R a relação no conjunto de pares ordenados de inteiros positivos tal que $((a, b), (c, d)) \in R$ se e somente se $a + d = b + c$. Mostre que R é uma relação de equivalência.

16. Seja R a relação no conjunto de pares ordenados de inteiros positivos tal que $((a, b), (c, d)) \in R$ se e somente se $ad = bc$. Mostre que R é uma relação de equivalência.

17. (*Requer cálculo*)
a) Mostre que a relação R no conjunto de todas as funções diferenciáveis de **R** para **R** que consiste em todos os pares (f, g) tal que $f'(x) = g'(x)$ para qualquer número real x é uma relação de equivalência.
b) Quais funções estão na mesma classe de equivalência que a função $f(x) = x^2$?

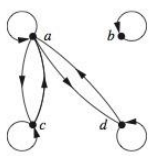
18. (*Requer cálculo*)
a) Seja n um inteiro positivo. Mostre que a relação R no conjunto de todos os polinômios com coeficientes reais, que consiste em todos os pares (f, g) tal que $f^{(n)}(x) = g^{(n)}(x)$, é uma relação de equivalência. [Aqui, $f^{(n)}(x)$ é a enésima derivada de $f(x)$.]
b) Quais funções estão na mesma classe de equivalência que a função $f(x) = x^4$, em que $n = 3$?

19. Seja R a relação no conjunto de todas as URLs (ou endereços na Web) tal que $x\,R\,y$ se e somente se a página na Web em x é a mesma que a página na Web em y. Mostre que R é uma relação de equivalência.

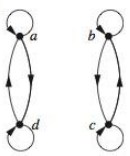
20. Seja R a relação no conjunto de todas as pessoas que já visitaram uma página particular na Web tal que $x\,R\,y$ se e somente se a pessoa x e a pessoa y seguiram o mesmo conjunto de links, começando nessa página da Web (indo de página da Web para página da Web até que parassem de usar a Web). Mostre que R é uma relação de equivalência.

Nos exercícios 21 a 23, determine se a relação cujo grafo orientado é mostrado é uma relação de equivalência.

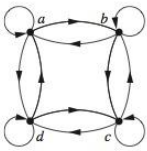
21.



22.



23.



24. Determine se as relações representadas por estas matrizes zero-um são relações de equivalência.

a) $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$ **c)** $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

25. Mostre que a relação R no conjunto de todas as seqüências de bits tal que $s\ R\ t$ se e somente se s e t contiverem o mesmo número de 1s é uma relação de equivalência.

26. Quais são as classes de equivalência das relações de equivalência do Exercício 1?

27. Quais são as classes de equivalência das relações de equivalência do Exercício 2?

28. Quais são as classes de equivalência das relações de equivalência do Exercício 3?

29. Quais são as classes de equivalência da seqüência de bits 011 para a relação de equivalência do Exercício 25?

30. Quais são as classes de equivalência destas seqüências de bits para a relação de equivalência do Exercício 11?

a) 010 **b)** 1011 **c)** 11111 **d)** 01010101

31. Quais são as classes de equivalência das seqüências de bits do Exercício 30 para as relações de equivalência do Exercício 11?

32. Quais são as classes de equivalência das seqüências de bits do Exercício 30 para as relações de equivalência do Exercício 12?

33. Quais são as classes de equivalência das seqüências de bits do Exercício 30 para a relação de equivalência R_4 do Exemplo 5, no conjunto de todas as seqüências de bits? (Lembre-se que as seqüências de bits s e t são equivalentes segundo R_4 se e somente se elas forem iguais ou ambas tiverem, pelo menos, 4 bits de comprimento e coincidirem em seus primeiros 4 bits.)

34. Quais são as classes de equivalência das seqüências de bits do Exercício 30 para a relação de equivalência R_5 do Exemplo 5, no conjunto de todas as seqüências de bits? (Lembre-se de que as seqüências de bits s e t são equivalentes segundo R_5 se e somente se elas forem iguais ou ambas tiverem, pelo menos, 5 bits de comprimento e coincidirem em seus primeiros 5 bits.)

35. Qual é a classe de congruência $[n]_5$ (ou seja, a classe de equivalência de n relativa à congruência módulo 5) quando n for

a) 2? **b)** 3? **c)** 6? **d)** -3?

36. Qual é a classe de congruência $[4]_m$ quando m for

a) 2? **b)** 3? **c)** 6? **d)** 8?

37. Dê uma descrição de cada classe de congruência módulo 6.

38. Qual é a classe de equivalência das seqüências relativas à relação de equivalência do Exercício 14?

a) *No* **b)** *Yes* **c)** *Help*

39. **a)** Qual é a classe de equivalência de (1, 2) relativa à relação de equivalência do Exercício 15?

b) Dê uma interpretação das classes de equivalência para a relação de equivalência R do Exercício 15. [*Dica*: Observe a diferença $a - b$ correspondente a (a, b).]

40. **a)** Qual é a classe de equivalência de (1, 2) relativa à relação de equivalência do Exercício 16?

b) Dê uma interpretação das classes de equivalência para a relação de equivalência R do Exercício 16. [*Dica*: Observe a razão a/b correspondente a (a, b).]

41. Quais destas coleções de subconjuntos são partições de $\{1, 2, 3, 4, 5, 6\}$?

a) $\{1, 2\}, \{2, 3, 4\}, \{4, 5, 6\}$ **b)** $\{1,\}, \{2, 3, 6\}, \{4\}, \{5\}$

c) $\{2, 4, 6\}, \{1, 3, 5\}$ **d)** $\{1, 4, 5\}, \{2, 6\}$

42. Quais destas coleções de subconjuntos são partições de $\{-3, -2, -1, 0, 1, 2, 3\}$?

a) $\{-3, -1, 1, 3\}, \{-2, 0, 2\}$

b) $[-3, -2, -1, 0\}, \{0, 1, 2, 3\}$

c) $\{-3, 3\}, \{-2, 2\}, \{-1, 1\}, \{0\}$

d) $\{-3, -2, 2, 3\}, \{-1, 1\}$

43. Quais destas coleções de subconjuntos são partições no conjunto de seqüências de bits de comprimento 8?

a) o conjunto das seqüências de bits que começam com 1, o conjunto das seqüências de bits que começam com 00 e o conjunto das seqüências de bits que começam com 01

b) o conjunto das seqüências de bits que contêm a seqüência 00, o conjunto das seqüências de bits que contêm a seqüência 01, o conjunto das seqüências de bits que contêm a seqüência 10 e o conjunto das seqüências de bits que contêm a seqüência 11

c) o conjunto das seqüências de bits que terminam com 00, o conjunto das seqüências de bits que terminam com 01, o conjunto das seqüências de bits que terminam com 10 e o conjunto das seqüências de bits que terminam com 11

d) o conjunto das seqüências de bits que terminam com 111, o conjunto das seqüências de bits que terminam com 011 e o conjunto das seqüências de bits que terminam com 00

e) o conjunto das seqüências de bits que têm $3k$ uns, em que k é um inteiro não negativo; o conjunto das seqüências de bits que têm $3k + 1$ uns, em que k é um inteiro não negativo; e o conjunto das seqüências de bits que têm $3k + 2$ uns, em que k é um inteiro não negativo

44. Quais destas coleções de subconjuntos são partições do conjunto dos inteiros?

a) o conjunto dos inteiros pares e o conjunto dos inteiros ímpares

b) o conjunto dos inteiros positivos e o conjunto dos inteiros negativos

c) o conjunto dos inteiros divisíveis por 3, o conjunto dos inteiros que deixam resto 1 quando divididos por 3 e o conjunto dos inteiros que deixam resto 2 quando divididos por 3

d) o conjunto dos inteiros menores que -100, o conjunto dos inteiros cujo valor absoluto não exceda 100 e o conjunto dos inteiros maiores que 100

e) o conjunto dos inteiros que não são divisíveis por 3, o conjunto dos inteiros pares e o conjunto dos inteiros que deixam resto 3 quando divididos por 6

45. Quais destes são partições do conjunto $\mathbf{Z} \times \mathbf{Z}$ de pares ordenados de inteiros?

a) o conjunto dos pares (x, y), no qual x ou y é ímpar; o conjunto dos pares (x, y), no qual x é par; e o conjunto dos pares (x, y), no qual y é par

b) o conjunto dos pares (x, y), no qual x e y são ambos ímpares; o conjunto dos pares (x, y), no qual exatamente um entre x e y é ímpar; e o conjunto dos pares (x, y), no qual x e y são ambos pares

c) o conjunto dos pares (x, y), no qual x é positivo; o conjunto dos pares (x, y), no qual y é positivo; e o conjunto dos pares (x, y), no qual x e y são ambos negativos

d) o conjunto dos pares (x, y), no qual $3 \mid x$ e $3 \mid y$; o conjunto dos pares (x, y), no qual $3 \mid x$ e $3 \nmid y$; o conjunto dos pares (x, y), no qual $3 \nmid x$ e $3 \mid y$; e o conjunto dos pares (x, y), no qual $3 \nmid x$ e $3 \nmid y$

e) o conjunto dos pares (x, y), no qual $x > 0$ e $y > 0$; o conjunto dos pares (x, y), no qual $x > 0$ e $y \leq 0$; o conjunto dos pares (x, y), no qual $x \leq 0$ e $y > 0$; e o conjunto dos pares (x, y), no qual $x \leq 0$ e $y \leq 0$

f) o conjunto dos pares (x, y), no qual $x \neq 0$ e $y \neq 0$; o conjunto dos pares (x, y), no qual $x = 0$ e $y \neq 0$; e o conjunto dos pares (x, y), no qual $x \neq 0$ e $y = 0$

46. Quais destes são partições do conjunto dos números reais?

a) os números reais negativos, $\{0\}$, os números reais positivos

b) o conjunto dos números irracionais, o conjunto dos números racionais

c) o conjunto de intervalos $[k, k + 1]$, $k = \ldots, -2, -1, 0, 1, 2, \ldots$

d) o conjunto de intervalos $(k, k + 1)$, $k = \ldots, -2, -1, 0, 1, 2, \ldots$

e) o conjunto de intervalos $(k, k + 1]$, $k = \ldots, -2, -1, 0, 1, 2, \ldots$

f) os conjuntos $\{x + n \mid n \in \mathbf{Z}\}$, para todo $x \in [0, 1)$

47. Liste os pares ordenados nas relações de equivalência produzidas por estas partições de $\{0, 1, 2, 3, 4, 5\}$.

a) $\{0\}, \{1, 2\}, \{3, 4, 5\}$

b) $\{0, 1\}\ \{2, 3\}, \{4, 5\}$

c) $\{0, 1, 2\}, \{3, 4, 5\}$

d) $\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}$

48. Liste os pares ordenados nas relações de equivalência produzidas por estas partições de $\{a, b, c, d, e, f, g\}$.

a) $\{a, b\}, \{c, d\}, \{e, f, g\}$

b) $\{a\}, \{b\}, \{c, d\}, \{e, f\}, \{g\}$

c) $\{a, b, c, d\}\ \{e, f, g\}$

d) $\{a, c, e, g\}, \{b, d\}, \{f\}$

Uma partição P_1 é denominada um **refinamento** da partição P_2 se todo conjunto em P_1 for um subconjunto de um dos conjuntos em P_2.

49. Mostre que a partição formada a partir das classes de congruência módulo 6 é um refinamento da partição obtida a partir das classes de congruência módulo 3.

50. Mostre que a partição do conjunto de pessoas que vivem nos Estados Unidos, que consiste nos subconjuntos de pessoas que vivem no mesmo distrito (ou paróquia) e no mesmo estado, é um refinamento da partição que consiste nos subconjuntos das pessoas que vivem no mesmo estado.

51. Mostre que a partição do conjunto de seqüências de bits de comprimento 16, formada pelas classes de equivalência de seqüências de bits que coincidem nos últimos 8 bits, é um refinamento da partição formada a partir das classes de equivalência de seqüências de bits que coincidem nos últimos 4 bits.

Nos exercícios 52 e 53, R_n se refere à família de relações de equivalência definida no Exemplo 5. Lembre-se de que $s\ R_n\ t$, em que s e t são duas seqüências se $s = t$ ou se s e t forem seqüências com, pelo menos, n caracteres que coincidam em seus primeiros n caracteres.

52. Mostre que a partição do conjunto de todas as seqüências de bits, formada pelas classes de equivalência de seqüências de bits relativas à relação de equivalência R_4, é um refinamento da partição formada pelas classes de equivalência de seqüências de bits relativas à relação de equivalência R_3.

53. Mostre que a partição do conjunto de todos os identificadores em C, formada pelas classes de equivalência de identificadores relativas à relação de equivalência R_{31}, é um refinamento da partição formada pelas classes de equivalência de identificadores relativas à relação de equivalência R_8. (Os compiladores para o "velho" C consideravam identificadores como o mesmo quando seus nomes coincidiam até os oito primeiros caracteres, enquanto os compiladores no standard C consideram identificadores como o mesmo quando seus nomes coincidem até os 31 primeiros caracteres.)

54. Suponha que R_1 e R_2 sejam relações de equivalência em um conjunto A. Sejam P_1 e P_2 as partições correspondentes a R_1 e R_2, respectivamente. Mostre que $R_1 \subseteq R_2$ se e somente se P_1 for um refinamento de P_2.

55. Encontre a menor relação de equivalência no conjunto $\{a, b, c, d, e\}$ que contenha a relação $\{(a, b), (a, c), (d, e)\}$.

56. Suponha que R_1 e R_2 sejam relações de equivalência em um conjunto S. Determine se cada uma destas combinações de R_1 e R_2 deve ser uma relação de equivalência.

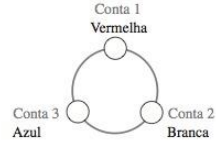
a) $R_1 \cup R_2$ b) $R_1 \cap R_2$ $R_1 \oplus R_2$

57. Considere a relação de equivalência do Exemplo 3, ou seja, $R = \{(x, y) \mid x - y \text{ é um inteiro}\}$.

a) Qual é a classe de equivalência de 1 para essa relação de equivalência?

b) Qual é a classe de equivalência de 1/2 para essa relação de equivalência?

***58.** Cada conta em um bracelete com 3 contas é vermelha, branca ou azul, como ilustrado na figura.



Defina a relação R entre braceletes como: (B_1, B_2), em que B_1 e B_2 são braceletes, pertence a R se e somente se B_2 puder ser obtido de B_1 por uma rotação ou por uma rotação seguida de uma reflexão.

a) Mostre que R é uma relação de equivalência.
b) Quais são as classes de equivalência de R?

***59.** Seja R a relação no conjunto de todas as colorações possíveis de tabuleiros 2×2, nos quais cada um de seus quatro quadrados está pintado ou de vermelho ou de azul, de modo que (C_1, C_2), em que C_1 e C_2 são tabuleiros 2×2, com cada um de seus quatro quadrados pintado de vermelho ou azul, pertence a R se e somente se C_2 puder ser obtido a partir de C_1 ou girando o tabuleiro ou girando-o e, a seguir, refletindo-o.
a) Mostre que R é uma relação de equivalência.
b) Quais são as classes de equivalência de R?

60. **a)** Seja R a relação no conjunto de funções de $\mathbf{Z}^+$ para $\mathbf{Z}^+$ tal que (f, g) pertence a R se e somente se f for $\Theta(g)$ (ver Seção 3.2). Mostre que R é uma relação de equivalência.
b) Descreva a classe de equivalência que contém $f(n) = n^2$ para a relação de equivalência da parte (a).

61. Determine o número de relações de equivalência diferentes em um conjunto com três elementos listando-as.

62. Determine o número de relações de equivalência diferentes em um conjunto com quatro elementos listando-as.

***63.** Nós necessariamente obtemos uma relação de equivalência quando formamos o fecho transitivo do fecho simétrico do fecho reflexivo de uma relação?

***64.** Nós necessariamente obtemos uma relação de equivalência quando formamos o fecho simétrico do fecho reflexivo do fecho transitivo de uma relação?

65. Suponha que usemos o Teorema 2 para formar uma partição P a partir de uma relação de equivalência R. Qual é a relação de equivalência R' que resulta se usarmos novamente o Teorema 2 para formar uma relação de equivalência a partir de P?

66. Suponha que usemos o Teorema 2 para formar uma relação de equivalência R a partir de uma partição P. Qual é a partição P' que resulta se usarmos novamente o Teorema 2 para formar uma partição a partir de R?

67. Invente um algoritmo para encontrar a menor relação de equivalência que contenha uma dada relação.

***68.** Seja $p(n)$ o número de relações de equivalência diferentes em um conjunto com n elementos (pelo Teorema 2, o número de partições de um conjunto com n elementos). Mostre que $p(n)$ satisfaz a relação de recorrência $p(n) = \sum_{j=0}^{n-1} C(n-1, j)p(n-j-1)$ e a condição inicial $p(0) = 1$. (*Observação*: Os números $p(n)$ são chamados de **números de Bell** em homenagem ao matemático americano E. T. Bell.)

69. Use o Exercício 68 para encontrar o número de relações de equivalência diferentes em um conjunto com n elementos, em que n é um inteiro positivo que não excede 10.

8.6 Ordens Parciais

Introdução

Com freqüência usamos relações para ordenar alguns ou todos os elementos de conjuntos. Por exemplo, ordenamos palavras usando a relação que contém pares de palavras (x, y), em que x vem antes de y no dicionário. Nós fazemos o planejamento de projetos usando a relação que consiste nos pares (x, y), em que x e y são tarefas em um projeto tal que x deve ser completada antes do início de y. Ordenamos o conjunto dos inteiros usando a relação que contém os pares (x, y), em que x é menor que y. Quando adicionamos todos os pares da forma (x, x) a estas relações, obtemos uma relação que é reflexiva, anti-simétrica e transitiva. Estas são as propriedades que caracterizam relações usadas para ordenar os elementos de conjuntos.

Links

DEFINIÇÃO 1 Uma relação R em um conjunto S é chamada de *ordenação parcial* ou de *ordem parcial* se for reflexiva, anti-simétrica e transitiva. Um conjunto S junto com uma ordenação parcial R é denominado um *conjunto parcialmente ordenado*, ou *poset* (de *partially ordered set*, em inglês) e é indicado por (S, R). Os membros de S são chamados de *elementos* do poset.

Apresentamos posets nos exemplos 1 a 3.

EXEMPLO 1 Mostre que a relação "maior que ou igual a" ($\geq$) é uma ordem parcial no conjunto dos inteiros.

Exemplos Extras

Solução: Como $a \geq a$ para todo inteiro a, $\geq$ é reflexiva. Se $a \geq b$ e $b \geq a$, então $a = b$. Logo, $\geq$ é anti-simétrica. Finalmente, $\geq$ é transitiva, pois $a \geq b$ e $b \geq c$ implicam que $a \geq c$. Segue que $\geq$ é uma ordem parcial no conjunto dos inteiros e $(\mathbf{Z}, \geq)$ é um poset. ◄

EXEMPLO 2 A relação de divisibilidade $|$ é uma ordem parcial no conjunto dos inteiros positivos, pois ela é reflexiva, anti-simétrica e transitiva, como foi mostrado na Seção 8.1. Vemos que $(\mathbf{Z}^+, |)$ é um poset. Lembre-se de que ($\mathbf{Z}^+$ indica o conjunto dos inteiros positivos.) ◀

EXEMPLO 3 Mostre que a relação de inclusão $\subseteq$ é uma ordem parcial no conjunto das partes de um conjunto S.

Solução: Como $A \subseteq A$ sempre que A for um subconjunto de S, $\subseteq$ é reflexiva. É anti-simétrica, pois $A \subseteq B$ e $B \subseteq A$ implicam que $A = B$. Finalmente, $\subseteq$ é transitiva, pois $A \subseteq B$ e $B \subseteq C$ implicam que $A \subseteq C$. Logo, $\subseteq$ é uma ordem parcial em $P(S)$ e $(P(S), \subseteq)$ é um poset. ◀

O Exemplo 4 ilustra uma relação que não é uma ordem parcial.

EXEMPLO 4 Seja R a relação no conjunto das pessoas tal que $x\,R\,y$ se x e y forem pessoas e x for mais velha do que y. Mostre que R não é uma ordem parcial.

Exemplos Extras

Solução: Observe que R é anti-simétrica porque se uma pessoa x for mais velha do que uma pessoa y , então y não é mais velha do que x. Isto é, xRy, então $y \not R x$. A relação R é transitiva porque se uma pessoa x for mais velha do que uma pessoa y e y for mais velha do que uma pessoa z, então x é mais velha do que z. Isto é, se xRy e yRz, então xRz. Entretanto, R não é reflexiva, pois nenhuma pessoa é mais velha do que si mesma. Isto é, $x \not R x$ para qualquer pessoa x. Segue que R não é uma ordem parcial. ◀

Em posets diferentes são usados símbolos diferentes, tais como $\leq$, $\subseteq$ e $|$, para uma ordem parcial. Entretanto, precisamos de um símbolo que possa ser usado quando discutirmos a relação de ordem em um poset arbitrário. Em geral, a indicação $a \preccurlyeq b$ é usada para denotar que $(a, b) \in R$ em um poset arbitrário (S, R). Esta notação é usada porque a relação "menor que ou igual a" no conjunto dos números reais é o exemplo mais familiar de ordem parcial e o símbolo $\preccurlyeq$ é parecido com o símbolo $\leq$. (Observe que o símbolo $\preccurlyeq$ é usado para indicar a relação em *qualquer* poset, e não apenas a relação "menor que ou igual a".) A notação $a \prec b$ significa que $a \preccurlyeq b$, mas $a \neq b$. Além disso, dizemos que "a é menor que b" ou "b é maior que a" se $a \prec b$.

Quando a e b forem elementos do poset $(S, \preccurlyeq)$, não é necessário nem que $a \preccurlyeq b$ nem que $b \preccurlyeq a$. Por exemplo, em $(P(\mathbf{Z}), \subseteq)$, $\{1, 2\}$ não está relacionado com $\{1, 3\}$ e vice-versa, pois nenhum dos dois conjuntos está contido no outro. Analogamente, em $(\mathbf{Z}^+, |)$, 2 não está relacionado com 3 e 3 não está relacionado com 2, pois $2 \not| 3$ e $3 \not| 2$. Isto leva à Definição 2.

DEFINIÇÃO 2 Os elementos a e b de um poset $(S, \preccurlyeq)$ são denominados *comparáveis* se ou $a \preccurlyeq b$ ou $b \preccurlyeq a$. Quando a e b forem elementos de S tal que nem $a \preccurlyeq b$ nem $b \preccurlyeq a$, a e b serão denominados *incomparáveis*.

EXEMPLO 5 No poset $(\mathbf{Z}^+, |)$, os inteiros 3 e 9 são comparáveis? 5 e 7 são comparáveis?

Solução: Os inteiros 3 e 9 são comparáveis, pois $3 \mid 9$. Os inteiros 5 e 7 são incomparáveis, pois $5 \not| 7$ e $7 \not| 5$. ◀

O adjetivo "parcial" é usado para descrever as ordens parciais porque pares de elementos podem ser incomparáveis. Quando quaisquer dois elementos no conjunto forem comparáveis, a relação é chamada de **ordem total**.

DEFINIÇÃO 3 Se $(S, \preccurlyeq)$ for um poset e quaisquer dois elementos de S forem comparáveis, S será chamado de *conjunto totalmente ordenado* ou *linearmente ordenado*, e $\preccurlyeq$ será chamada de *ordem total* ou *ordem linear*. Um conjunto totalmente ordenado também é chamado de *cadeia*.

EXEMPLO 6 O poset $(\mathbf{Z}, \leq)$ é totalmente ordenado, pois $a \leq b$ ou $b \leq a$ sempre que a e b forem inteiros. ◀

EXEMPLO 7 O poset $(\mathbf{Z}^+, |)$ não é totalmente ordenado, pois contém elementos que são incomparáveis, como 5 e 7. ◀

No Capítulo 4, observamos que $(\mathbf{Z}^+, \leq)$ é bem-ordenado, em que $\leq$ é a relação usual "menor que ou igual a". Definimos agora conjuntos bem-ordenados.

DEFINIÇÃO 4 $(S, \preccurlyeq)$ é um *conjunto bem-ordenado* se for um poset tal que $\preccurlyeq$ seja uma ordem total e todo subconjunto não vazio de S tenha um menor elemento.

EXEMPLO 8 O conjunto de pares ordenados de inteiros positivos, $\mathbf{Z}^+ \times \mathbf{Z}^+$, com $(a_1, a_2) \preccurlyeq (b_1, b_2)$ se $a_1 < b_1$ ou se $a_1 = b_1$ e $a_2 \leq b_2$ (a ordem lexicográfica), é um conjunto bem-ordenado. A verificação disso é deixada como exercício no final desta seção. O conjunto $\mathbf{Z}$, com a ordem usual $\leq$, não é bem-ordenado, pois o conjunto dos inteiros negativos, que é um subconjunto de $\mathbf{Z}$, não tem um menor elemento. ◀

Na Seção 4.3, mostramos como usar o princípio de indução da boa ordem (lá chamado de indução generalizada) para demonstrar resultados sobre conjuntos bem-ordenados. Nós agora enunciaremos e demonstraremos que esta técnica de demonstração é válida.

TEOREMA 1 **O PRINCÍPIO DE INDUÇÃO DA BOA ORDEM** Suponha que S seja um conjunto bem-ordenado. Então, $P(x)$ será verdadeira para todo $x \in S$ se

PASSO DE INDUÇÃO: Para todo $y \in S$, se $P(x)$ for verdadeira para todo $x \in S$, com $x \prec y$, então $P(y)$ será verdadeira.

***Demonstração*:** Suponha que não seja o caso de $P(x)$ ser verdadeira para todo $x \in S$. Então, existe um elemento $y \in S$ tal que $P(y)$ é falsa. Conseqüentemente, o conjunto $A = \{x \in S \mid P(x)$ é falso$\}$ é não vazio. Como S é bem-ordenado, A tem um menor elemento a. Pela escolha de a como menor elemento de A, sabemos que $P(x)$ é verdadeira para todo $x \in S$ com $x \prec a$. Isso implica que o passo de indução $P(a)$ é verdadeiro. Esta contradição mostra que $P(x)$ deve ser verdadeira para todo $x \in S$. ◁

***Lembre-se*:** Nós não precisamos de um passo base na demonstração que usa o princípio da indução da boa ordem, porque se x_0 for o menor elemento de um conjunto bem-ordenado, o passo de indução nos diz que $P(x_0)$ é verdadeira. Isso ocorre porque não existe nenhum elemento $x \in S$ com $x \prec x_0$, de modo que sabemos (usando uma demonstração por vacuidade) que $P(x)$ é verdadeira para todo $x \in S$ com $x \prec x_0$.

O princípio da indução da boa ordem é uma técnica versátil para demonstrar resultados sobre conjuntos bem-ordenados. Mesmo quando é possível usar a indução matemática para o conjunto dos inteiros positivos para demonstrar um teorema, pode ser mais simples usar o princípio da

indução da boa ordem, como vimos nos exemplos 5 e 6 da Seção 4.2, nos quais demonstramos um resultado sobre o conjunto bem-ordenado $(\mathbf{N} \times \mathbf{N}, \preccurlyeq)$, em que $\preccurlyeq$ é a ordem lexicográfica em $\mathbf{N} \times \mathbf{N}$.

Ordem Lexicográfica

As palavras em um dicionário estão listadas em ordem alfabética, ou lexicográfica, que se baseia na ordem das letras no alfabeto. Esse é um caso especial de ordem de seqüências em um conjunto construída a partir de uma ordem parcial no conjunto. Vamos ver como essa construção funciona em qualquer poset.

Inicialmente, mostraremos como construir uma ordem parcial no produto cartesiano de dois posets, $(A_1, \preccurlyeq_1)$ e $(A_2, \preccurlyeq_2)$. A **ordem lexicográfica** $\preccurlyeq$ em $A_1 \times A_2$ é definida especificando que um par é menor que um segundo par se o primeiro elemento do primeiro par for menor do que o primeiro elemento do segundo par (em A_1), ou se os primeiros elementos forem iguais, mas o segundo elemento deste par for menor do que o segundo elemento do segundo par (em A_2). Em outras palavras, (a_1, a_2) é menor que (b_1, b_2), ou seja,

$$(a_1, a_2) \prec (b_1, b_2),$$

ou se $a_1 \prec_1 b_1$ ou se ambos $a_1 = b_1$ e $a_2 \prec_2 b_2$.

Obtemos uma ordem parcial $\preccurlyeq$ adicionando a igualdade à ordem $\prec$ em $A_1 \times A_2$. A verificação disso é deixada como exercício.

EXEMPLO 9 Determine se $(3, 5) \prec (4, 8)$, se $(3, 8) \prec (4, 5)$ e se $(4, 9) \prec (4, 11)$ no poset $(\mathbf{Z} \times \mathbf{Z}, \preccurlyeq)$, em que $\preccurlyeq$ é a ordem lexicográfica construída a partir da relação $\leq$ usual em $\mathbf{Z}$.

Solução: Como $3 < 4$, segue que $(3, 5) \prec (4, 8)$ e $(3, 8) \prec (4, 5)$. Temos $(4, 9) \prec (4, 11)$, porque os primeiros elementos de $(4, 9)$ e $(4, 11)$ são os mesmos, mas $9 < 11$. ◀

Na Figura 1, os pares ordenados em $\mathbf{Z}^+ \times \mathbf{Z}^+$ que são menores que $(3, 4)$ estão destacados. Pode ser definida uma ordem lexicográfica no produto cartesiano de n posets $(A_1, \preccurlyeq_1)$, $(A_2, \preccurlyeq_2)$, …, $(A_n, \preccurlyeq_n)$. Defina a ordem parcial $\preccurlyeq$ em $A_1 \times A_2 \times \cdots \times A_n$ por

$$(a_1, a_2, \ldots, a_n) \prec (b_1, b_2, \ldots, b_n)$$

se $a_1 \prec_1 b_1$, ou se existir um inteiro $i > 0$ tal que $a_1 = b_1, \ldots, a_i = b_1$, e $a_{i+1} \prec_{i+1} b_{i+1}$. Em outras palavras, uma n-upla será menor do que uma segunda n-upla se o elemento da primeira n-upla na primeira posição, na qual as duas n-uplas não coincidem, for menor do que o elemento naquela posição na segunda n-upla.

EXEMPLO 10 Observe que $(1, 2, 3, 5) \prec (1, 2, 4, 3)$, pois os elementos nas primeiras duas posições destas 4-uplas coincidem, mas na terceira posição o elemento na primeira 4-upla, 3, é menor que o da segunda 4-upla, 4. (Aqui, a ordem das 4-uplas é a ordem lexicográfica que vem da relação "menor que ou igual a" usual no conjunto dos inteiros.) ◀

Podemos agora definir a ordem lexicográfica de seqüências. Considere as seqüências $a_1a_2 \ldots a_m$ e $b_1b_2 \ldots b_n$ em um conjunto parcialmente ordenado S. Suponha que essas seqüências não sejam iguais. Seja t o mínimo de m e n. A definição da ordem lexicográfica é que a seqüência $a_1a_2 \ldots a_m$ é menor que

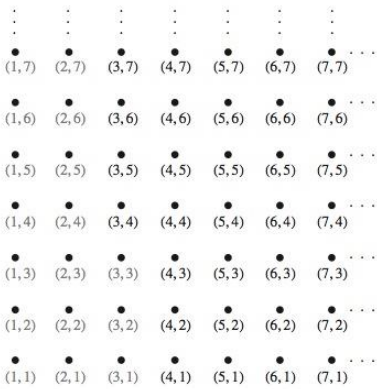


FIGURA 1 Os Pares Ordenados Menores que (3, 4) na Ordem Lexicográfica.

$b_1b_2 \ldots b_n$ se e somente se

$$(a_1, a_2, \ldots, a_t) \prec (b_1, b_2, \ldots, b_t), \text{ ou}$$
$$(a_1, a_2, \ldots, a_t) = (b_1, b_2, \ldots, b_t) \text{ e } m < n,$$

em que $\prec$ nesta desigualdade representa a ordem lexicográfica de S^*. Em outras palavras, para determinar a ordem de duas seqüências diferentes, a seqüência mais longa é truncada para o comprimento da seqüência mais curta, ou seja, $t = \min(m, n)$ termos. Então, as t-uplas formadas dos primeiros t termos de cada seqüência são comparadas usando a ordem lexicográfica em S^*. Uma seqüência é menor que outra seqüência se a t-upla correspondente à primeira seqüência for menor do que a t-upla correspondente à segunda seqüência ou se estas duas t-uplas forem as mesmas, mas a segunda seqüência for mais longa. A verificação de que esta é uma ordem parcial é deixada como exercício para o leitor.

EXEMPLO 11 Considere o conjunto das seqüências de letras minúsculas da língua inglesa. Usando a ordem das letras no alfabeto, pode ser construída uma ordem lexicográfica no conjunto das seqüências. Uma seqüência é menor do que uma segunda seqüência se a letra na primeira seqüência na primeira posição, em que as seqüências sejam diferentes, vem antes da letra na segunda seqüência nesta posição, ou se a primeira seqüência e a segunda seqüência coincidem em todas as posições, mas a segunda seqüência tem mais letras. Essa ordem é a mesma usada nos dicionários. Por exemplo,

discreet $\prec$ *discrete*,

pois estas duas seqüências diferem pela primeira vez na sétima posição e $e \prec t$. Ainda,

discreet $\prec$ *discreetness*,

pois as primeiras oito letras coincidem, mas a segunda seqüência é mais longa. Além disso,

discrete $\prec$ *discretion*,

pois

discrete $\prec$ *discreti*. ◀

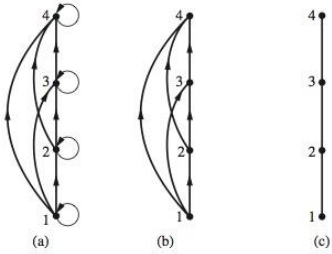


FIGURA 2 Construção de um Diagrama de Hasse para $(\{1, 2, 3, 4\}, \leq)$.

Diagrama de Hasse

Muitas arestas em um grafo orientado para um poset finito não precisam ser mostradas porque elas devem estar presentes. Por exemplo, considere o grafo orientado para a ordem parcial $\{(a, b) \mid a \leq b\}$ no conjunto $\{1, 2, 3, 4\}$, mostrado na Figura 2(a). Como esta relação é uma ordem parcial, ela é reflexiva e seu grafo orientado tem laços em todos os vértices. Conseqüentemente, não precisamos mostrar esses laços, pois eles devem estar presentes; na Figura 2(b), os laços não são mostrados. Como uma ordem parcial é transitiva, não temos de mostrar aquelas arestas que devem estar presentes por causa da transitividade. Por exemplo, na Figura 2(c) as arestas (1, 3), (1, 4) e (2, 4) não são mostradas porque elas devem estar presentes. Se supusermos que todas as arestas estão apontadas "para cima" (conforme foram desenhadas na figura), não temos de mostrar o sentido das arestas; a Figura 2(c) não mostra o sentido das arestas.

Em geral, podemos representar uma ordem parcial de um conjunto finito usando este procedimento: comece com o grafo orientado desta relação. Como uma ordem parcial é reflexiva, está presente um laço em cada vértice. Remova esses laços. Remova todas as arestas que devem estar presentes na ordem parcial por causa da presença de outras arestas e da transitividade. Por exemplo, se (a, b) e (b, c) estão na ordem parcial, remova a aresta (a, c), pois ela deve estar presente. Além disso, se (c, d) também estiver na ordem parcial, remova a aresta (a, d), pois ela também deve estar presente. Finalmente, arrume cada aresta de modo que seu vértice inicial esteja abaixo do seu vértice final (tal como está desenhada no papel). Remova todas as flechas no sentido das arestas, pois todas as arestas apontam "para cima" em direção a seu vértice final. (As arestas restantes correspondem aos pares na relação de recobrimento do poset. Ver o preâmbulo do Exercício 28.)

Estes passos estão bem definidos, e apenas um número finito de passos é necessário para um poset finito. Quando todos os passos tiverem sido executados, o diagrama resultante conterá informação suficiente para encontrar a ordem parcial. Este diagrama é chamado de **diagrama de Hasse** em homenagem ao matemático alemão do século XX, Helmut Hasse.

Links

HELMUT HASSE (1898–1979) Helmut Hasse nasceu em Kassel, na Alemanha. Ele serviu na marinha alemã depois do ensino médio. Começou seus estudos universitários na Universidade de Göttingen, em 1918, tendo se mudado para a Universidade de Marburg em 1920, para estudar com o especialista em teoria dos números Kurt Hensel. Durante essa época, Hasse fez contribuições fundamentais para a teoria algébrica dos números. Ele se tornou o sucessor de Hensel em Marburg, tendo mais tarde se tornado diretor do famoso instituto de matemática em Göttingen, em 1934, sendo contratado pela Universidade de Hamburgo em 1950. Hasse foi, por 50 anos, um editor do *Crelle's Journal*, uma famosa revista alemã de matemática, assumindo o papel de editor-chefe em 1936, quando os nazistas forçaram Hensel a renunciar. Durante a segunda grande guerra, Hasse trabalhou em pesquisa em matemática aplicada para a marinha alemã. Ele se destacou pela clareza e estilo pessoal de suas aulas e era dedicado tanto à teoria dos números quanto a seus alunos. (Hasse tem sido controverso por suas ligações com o partido nazista. Investigações mostraram que ele era um forte nacionalista alemão, mas não um nazista ardente.)

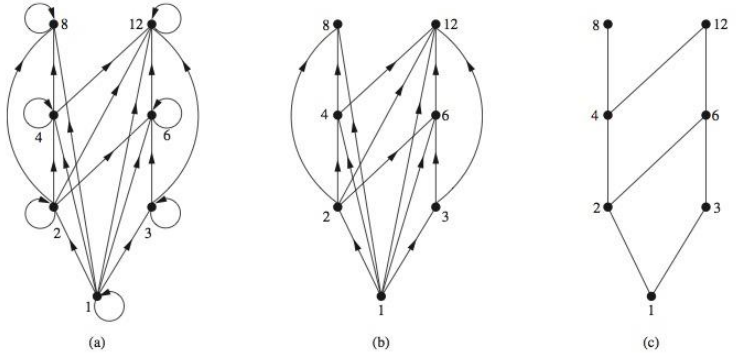


FIGURA 3 Construção do Diagrama de Hasse de ({1, 2, 3, 4, 6, 8, 12}, |).

EXEMPLO 12 Desenhe o diagrama de Hasse que representa a ordem parcial $\{(a, b) \mid a \text{ divide b}\}$ em {1, 2, 3, 4, 6, 8, 12}.

Solução: Comece com o dígrafo para esta ordem parcial, como mostrado na Figura 3(a). Remova todos os laços, como mostrado na Figura 3(b). A seguir, remova todas as arestas implicadas pela propriedade transitiva. Estas são (1, 4), (1, 6), (1, 8), (1, 12), (2, 8), (2, 12) e (3, 12). Arrume todas as arestas para apontarem para cima e apague todas as flechas para obter o diagrama de Hasse. O diagrama de Hasse resultante é mostrado na Figura 3(c). ◀

EXEMPLO 13 Desenhe o diagrama de Hasse para a ordem parcial $\{(A, B) \mid A \subseteq B\}$ no conjunto das partes $P(S)$, em que $S = \{a, b, c\}$.

Solução: O diagrama de Hasse para esta ordem parcial é obtido a partir do dígrafo associado apagando todos os laços e todas as arestas que ocorrem pela transitividade, ou seja, (∅, {a, b}), (∅, {a, c}), (∅, {b, c}), (∅, $\{a, b, c\}$), ($\{a\}$, $\{a, b, c\}$), ($\{b\}$, $\{a, b, c\}$) e ($\{c\}$, $\{a, b, c\}$). Finalmente, todas as arestas apontam para cima, e as flechas são apagadas. O diagrama de Hasse resultante está ilustrado na Figura 4. ◀

Elementos Maximais e Minimais

Elementos dos posets que têm determinadas propriedades extremais são importantes para muitas aplicações. Um elemento de um poset é chamado maximal se ele não for menor do que nenhum elemento do poset. Ou seja, a é **maximal** no poset $(S, \preccurlyeq)$ se não existir nenhum $b \in S$ tal que $a \prec b$. Analogamente, um elemento de um poset e chamado de minimal se ele não for maior do que nenhum elemento do poset. Ou seja, a é **minimal** se não existir nenhum elemento $b \in S$ tal que $b \prec a$. Elementos maximais e minimais são fáceis de descobrir usando um diagrama de Hasse. Eles são os elementos "de baixo" e "de cima" no diagrama.

EXEMPLO 14 Quais elementos do poset ({2, 4, 5, 10, 12, 20, 25}, |) são maximais e quais são minimais?

Solução: O diagrama de Hasse da Figura 5 para este poset mostra que os elementos maximais são 12, 20 e 25 e os elementos minimais são 2 e 5. Como esse exemplo mostra, um poset pode ter mais de um elemento maximal e mais de um elemento minimal. ◀

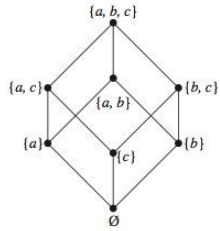


FIGURA 4 Diagrama de Hasse de ($P(\{a, b, c\})$, $\subseteq$).

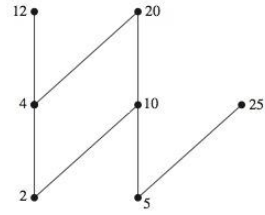


FIGURA 5 Diagrama de Hasse de um Poset.

Algumas vezes, existe um elemento em um poset que é maior do que todos os outros elementos. Esse elemento é chamado de maior elemento (ou máximo). Isto é, a é o **maior elemento** do poset $(S, \preccurlyeq)$ se $b \preccurlyeq a$ para todo $b \in S$. O maior elemento é único se existir [ver Exercício 40(a) no final desta seção]. Do mesmo modo, um elemento é chamado de menor elemento (ou mínimo) se ele for menor do que todos os outros elementos no poset. Ou seja, a é o **menor elemento** de $(S, \preccurlyeq)$ se $a \preccurlyeq b$ para todo $b \in S$. O menor elemento é único, se existir [ver Exercício 40(b) no final desta seção].

EXEMPLO 15 Determine se os posets representados em cada um dos diagramas de Hasse da Figura 6 têm um maior elemento e um menor elemento.

Solução: O menor elemento do poset com diagrama de Hasse (a) é a. Esse poset não tem maior elemento. O poset com diagrama de Hasse (b) não tem nem menor nem maior elemento. O poset com diagrama de Hasse (c) não tem menor elemento. Seu maior elemento é d. O poset com diagrama de Hasse (d) tem menor elemento a e maior elemento d. ◀

EXEMPLO 16 Seja S um conjunto. Determine se existe um menor elemento e um maior elemento no poset $(P(S), \subseteq)$.

Solução: O menor elemento é o conjunto vazio, pois $\emptyset \subseteq T$ para qualquer subconjunto T de S. O conjunto S é o maior elemento neste poset, pois $T \subseteq S$ sempre que T for um subconjunto de S. ◀

EXEMPLO 17 Existem um maior elemento e um menor elemento no poset $(\mathbf{Z}^+, |)$?

Solução: O inteiro 1 é o menor elemento, pois $1|n$ sempre que n for um inteiro positivo. Como não existe nenhum inteiro que seja divisível por todos os inteiros positivos, não existe o maior elemento. ◀

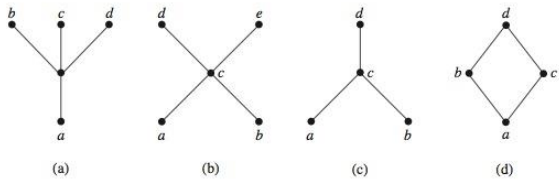


FIGURA 6 Diagramas de Hasse de Quatro Posets.

Algumas vezes, é possível encontrar um elemento que seja maior que ou igual a todos os elementos de um subconjunto A de um poset $(S, \preccurlyeq)$. Se u for um elemento de S tal que $a \preccurlyeq u$ para todos os elementos $a \in A$, então u é chamado de **limitante superior** de A. Do mesmo modo, pode existir um elemento que seja menor que ou igual a todos os elementos em A. Se l for um elemento de S tal que $l \preccurlyeq a$ para todos os elementos $a \in A$, então l é chamado um **limitante inferior** de A.

EXEMPLO 18 Encontre limitantes inferiores e superiores para os subconjuntos $\{a, b, c\}$, $\{j, h\}$ e $\{a, c, d, f\}$ no poset com diagrama de Hasse mostrado na Figura 7.

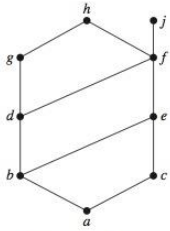


FIGURA 7 O Diagrama de Hasse de um Poset

Solução: Os limitantes superiores de $\{a, b, c\}$ são e, f, j e h, e seu único limitante inferior é a. Não existem limitantes superiores de $\{j, h\}$ e seus limitantes inferiores são a, b, c, d, e e f. Os limitantes superiores de $\{a, c, d, f\}$ são f, h, e j e seu limitante inferior é a. ◀

O elemento x é chamado de **menor limitante superior** do subconjunto A se x for um limitante superior que é menor do que qualquer outro limitante superior de A. Como existe apenas um elemento desse tipo, se existir, faz sentido chamar este elemento de o menor limitante superior [ver Exercício 42(a) no final desta seção]. Isto é, x é o menor limitante superior de A se $a \preccurlyeq x$ sempre que $a \in A$, e $x \preccurlyeq z$ sempre que z for um limitante superior de A. Analogamente, o elemento y é chamado de **maior limitante inferior** de A se y for um limitante inferior de A e $z \preccurlyeq y$ sempre que z for um limitante inferior de A. O maior limitante inferior de A é único, se existir [ver Exercício 42(b) no final desta seção]. O maior limitante inferior e o menor limitante superior de um subconjunto A são indicados por $\inf(A)$ e $\sup(A)$, respectivamente.

EXEMPLO 19 Encontre o maior limitante inferior e o menor limitante superior de $\{b, d, g\}$, se eles existirem, no poset mostrado na Figura 7.

Solução: Os limitantes superiores de $\{b, d, g\}$ são g e h. Como $g \prec h$, g é o menor limitante superior. Os limitantes inferiores de $\{b, d, g\}$ são a e b. Como $a \prec b$, b é o maior limitante inferior. ◀

EXEMPLO 20 Encontre o maior limitante inferior e o menor limitante superior dos conjuntos $\{3, 9, 12\}$ e $\{1, 2, 4, 5, 10\}$, se eles existirem, no poset $(\mathbf{Z}^+, |)$.

Exemplos Extras

Solução: Um inteiro é um limitante inferior de $\{3, 9, 12\}$ se 3, 9 e 12 forem divisíveis por esse inteiro. Os únicos tais inteiros são 1 e 3. Como $1 \mid 3$, 3 é o maior limitante inferior de $\{3, 9, 12\}$. O único limitante inferior do conjunto $\{1, 2, 4, 5, 10\}$ relativo a $|$ é o elemento 1. Logo, 1 é o maior limitante inferior de $\{1, 2, 4, 5, 10\}$.

Um inteiro é um limitante superior de $\{3, 9, 12\}$ se e somente se ele for divisível por 3, 9 e 12. Os inteiros com essa propriedade são aqueles divisíveis pelo mínimo múltiplo comum de 3, 9 e 12, que é 36. Logo, 36 é o menor limitante superior de $\{3, 9, 12\}$. Um inteiro positivo é um limitante superior de $\{1, 2, 4, 5, 10\}$ se e somente se ele for divisível por 1, 2, 4, 5 e 10. Os inteiros com essa propriedade são aqueles divisíveis pelo mínimo múltiplo comum desses inteiros, que é 20. Logo, 20 é o menor limitante superior de $\{1, 2, 4, 5, 10\}$. ◀

Reticulados

Um conjunto parcialmente ordenado no qual todo par de elementos tem tanto um menor limitante superior quanto um maior limitante inferior é chamado de **reticulado**. Os reticulados têm muitas propriedades especiais. Além disso, eles são usados em muitas aplicações diferentes, tais como modelos de fluxo de informações, e desempenham um papel importante na álgebra booleana.

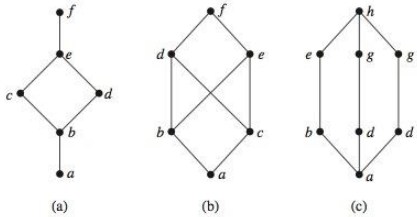


FIGURA 8 **Diagrama de Hasse de Três Posets.**

EXEMPLO 21 Determine se os posets representados pelos diagramas de Hasse na Figura 8 são reticulados.

Solução: Os posets representados pelos diagramas de Hasse em (a) e (c) são ambos reticulados, pois em cada um deles cada par de elementos tem tanto um maior limitante inferior quanto um menor limitante superior, como o leitor deve verificar. Por outro lado, o poset com diagrama de Hasse mostrado em (b) não é um reticulado, pois os elementos b e c não têm menor limitante superior. Para ver isso, observe que cada um dos elementos d, e e f é um limitante superior, mas nenhum desses três elementos precede os outros dois relativamente à ordem deste poset. ◀

EXEMPLO 22 O poset $(\mathbf{Z}^+, |)$ é um reticulado?

Solução: Sejam a e b dois inteiros positivos. O menor limitante superior e o maior limitante inferior desses dois inteiros são o mínimo múltiplo comum e o máximo divisor comum desses inteiros, respectivamente, como o leitor deve verificar. Segue que este poset é um reticulado. ◀

EXEMPLO 23 Determine se os posets $(\{1, 2, 3, 4, 5\}, |)$ e $(\{1, 2, 4, 8, 16\}, |)$ são reticulados.

Solução: Como 2 e 3 não têm limitantes superiores em $(\{1, 2, 3, 4, 5\}, |)$, eles certamente não têm um menor limitante superior. Logo, o primeiro poset não é um reticulado.

Todos os dois elementos do segundo poset têm tanto um menor limitante superior quanto um maior limitante inferior. O menor limitante superior de dois elementos deste poset é o maior dos elementos e o maior limitante inferior de dois elementos é o menor dos elementos, como o leitor deve verificar. Logo, este segundo poset é um reticulado. ◀

EXEMPLO 24 Determine se $(P(S), \subseteq)$, em que S é um conjunto, é um reticulado.

Solução: Sejam A e B dois subconjuntos de S. O menor limitante superior e o maior limitante inferior de A e B são $A \cup B$ e $A \cap B$, respectivamente, como o leitor pode verificar. Logo, $(P(S), \subseteq)$ é um reticulado. ◀

EXEMPLO 25 **O Modelo Reticulado do Fluxo de Informação** Em muitas situações, o fluxo de informação de uma pessoa, ou programa de computador, para outra é restrito por liberações de segurança. Podemos usar um modelo reticulado para representar diferentes políticas de fluxo de informação. Por exemplo, uma política de fluxo de informação comum é a *política de segurança multinível* usada em sistemas governamentais e militares. Cada parte da informação é designada para uma classe de segurança, e cada classe de segurança é representada por um par (A, C), em que A é um *nível de autoridade* e C é uma *categoria*. Então, é permitido que as pessoas e os programas de computador acessem informação de um conjunto restrito específico de classes de segurança.

Links

Os níveis de autoridade típicos usados pelo governo norte-americano são não classificado (0), confidencial (1), secreto (2) e supersecreto (3). As categorias usadas nas classes de segurança são os subconjuntos de um conjunto de todos os *compartimentos* relevantes para uma área parti-

cular de interesse. Cada compartimento representa uma área de assuntos particular. Por exemplo, se o conjunto dos compartimentos é {*espiões, informantes, agentes duplos*}, então existem oito categorias diferentes, uma para cada um dos oito subconjuntos do conjunto de compartimentos, tal como {*espiões, informantes*}.

Podemos coordenar as classes de segurança especificando que $(A_1, C_1) \preccurlyeq (A_2, C_2)$ se e somente se $A_1 \leq A_2$ e $C_1 \subseteq C_2$. É permitido que a informação flua da classe de segurança (A_1, C_1) para a classe de segurança (A_2, C_2) se e somente se $(A_1, C_1) \preccurlyeq (A_2, C_2)$. Por exemplo, é permitido que a informação flua da classe de segurança (*secreta*, {*espiões*, *informantes*}) para a classe de segurança (*supersecreta*, {*espiões*, *informantes*, *agentes duplos*}), enquanto não é permitido que a informação flua da classe de segurança (*supersecreto* {*espiões*, *informantes*}) para as classes de segurança (*secreto*, {*espiões*, *informantes*, *agentes duplos*}) ou (*supersecreto*, {*espiões*}).

Deixamos para o leitor (ver o Exercício 48 no final desta seção) mostrar que o conjunto de todas as classes de segurança com a ordem definida neste exemplo forma um reticulado. ◀

Ordenação Topológica

Links

Suponha que um projeto é feito de 20 tarefas diferentes. Algumas tarefas podem ser completadas apenas depois que outras tiverem terminado. Como encontrar uma ordem para essas tarefas? Para modelar este problema, montamos uma ordem parcial no conjunto das tarefas de modo que $a \prec b$ se e somente se a e b forem tarefas em que b não pode começar até que a tenha sido completada. Para produzir um planejamento para o projeto, precisamos produzir uma ordem para todas as 20 tarefas que seja compatível com esta ordem parcial. Mostraremos como isso pode ser feito.

Começamos com uma definição. Uma ordem total $\preccurlyeq$ é denominada **compatível** com a ordem parcial R se $a \preccurlyeq b$ sempre que aRb. Construir uma ordem total compatível a partir de uma ordem parcial é chamado de **ordenação topológica**.* Nós precisaremos usar o Lema 1.

LEMA 1 Todo poset não vazio finito $(S, \preccurlyeq)$ tem, pelo menos, um elemento minimal.

***Demostração*:** Escolha um elemento a_0 de S. Se a_0 não for minimal, então existe um elemento a_1, com $a_1 \prec a_0$. Se a_1 não for minimal, existe um elemento a_2, com $a_2 \prec a_1$. Continue este processo, de modo que, se a_n não for minimal, existe um elemento a_{n+1}, com $a_{n+1} \prec a_n$. Como existe apenas um número finito de elementos no poset, este processo deve terminar com um elemento minimal a_n. ◁

O algoritmo de ordenação topológica que descreveremos funciona para qualquer poset não vazio finito. Para definir uma ordem total no poset $(A, \preccurlyeq)$, primeiro escolhemos um elemento minimal a_1; esse elemento existe pelo Lema 1. A seguir, observe que $(A - \{a_1\}, \preccurlyeq)$ também é um poset, como o leitor deve verificar. (Aqui, por $\preccurlyeq$ queremos dizer a restrição da relação original $\preccurlyeq$ em A a $A - \{a_1\}$.) Se ele for não vazio, escolha um elemento minimal a_2 deste poset. Então remova a_2 também, e se ainda houver elementos adicionais, escolha um elemento minimal a_3 em $A - \{a_1, a_2\}$. Continue este processo escolhendo a_{k+1} como um elemento minimal de $A - \{a_1, a_2, \ldots, a_k\}$, enquanto sobrarem elementos.

Como A é um conjunto finito, este processo deve terminar. O produto final é uma seqüência de elementos $a_1, a_2, \ldots, a_n$. A ordem total $\preccurlyeq_t$ é definida por

$$a_1 \prec_t a_2 \prec_t \cdots \prec_t a_n.$$

* "Ordenação topológica" é uma terminologia usada pelos cientistas da computação, os matemáticos usam a terminologia "linearização de uma ordem parcial" para a mesma coisa. Em matemática, a topologia é um ramo da geometria que trata das propriedades das figuras geométricas que valem para todas as figuras que podem ser transformadas umas nas outras por bijeções contínuas. Em ciência da computação, uma topologia é qualquer arranjo de objetos que possam ser ligados por arestas.

Esta ordem total é compatível com a ordem parcial original. Para ver isso, observe que se $b \prec c$ na ordem parcial original, c é escolhido como o elemento minimal em uma fase do algoritmo em que b já foi removido, pois, caso contrário, c não seria um elemento minimal. O pseudocódigo para este algoritmo de ordenação topológica é mostrado no Algoritmo 1.

ALGORITMO 1 Ordenação Topológica

procedure *ordenacao topologica* ($(S, \preccurlyeq)$: poset finito)
$k := 1$
while $S \neq \emptyset$
begin
  $a_k :=$ um elemento minimal de S {esse elemento existe pelo Lema 1}
  $S := S - \{a_k\}$
  $k := k + 1$
end {$a_1, a_2, \ldots, a_n$ é uma ordem total compatível de S}

EXEMPLO 26 Encontre uma ordem total compatível para o poset $(\{1, 2, 4, 5, 12, 20\}, |)$.

Solução: O primeiro passo é escolher um elemento minimal. Este deve ser 1, pois é o único elemento minimal. A seguir, escolha um elemento minimal de $(\{2, 4, 5, 12, 20\}, |)$. Existem dois elementos minimais neste poset, a saber, 2 e 5. Escolhemos o 5. Os elementos restantes são {2, 4, 12, 20}. O único elemento minimal neste estágio é 2. A seguir, o 4 é escolhido porque ele é o único elemento minimal de $(\{4, 12, 20\}, |)$. Como tanto 12 quanto 20 são elementos minimais de $(\{12, 20\}, |)$, qualquer um deles pode ser escolhido a seguir. Escolhemos o 20, o que deixa o 12 como o último elemento. Isso produz a ordem total

$$1 \prec 5 \prec 2 \prec 4 \prec 20 \prec 12.$$

Os passos usados por este algoritmo de ordenação estão mostrados na Figura 9. ◀

A ordenação topológica tem uma aplicação no planejamento de projetos.

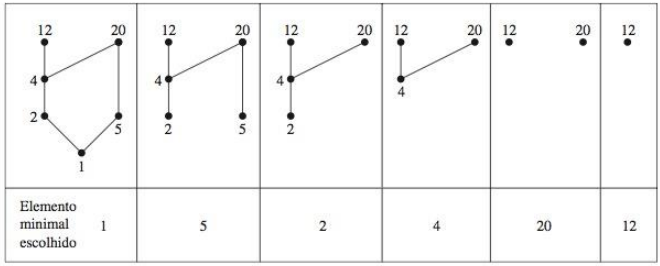


FIGURA 9 Uma Ordenação Topológica de $(\{1, 2, 4, 5, 12, 20\}, |)$.

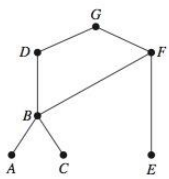


FIGURA 10 O Diagrama de Hasse para as Sete Tarefas.

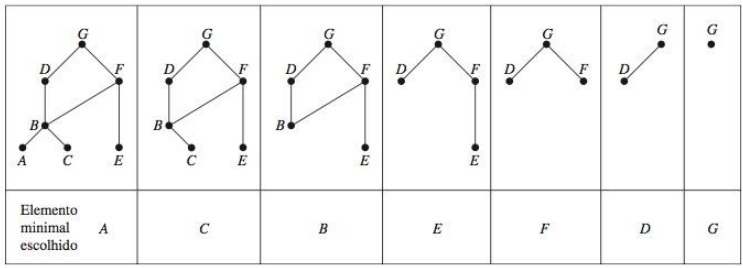


FIGURA 11 Uma Ordenação Topológica das Tarefas.

EXEMPLO 27 Um projeto de desenvolvimento em uma companhia de computação exige que se completem sete tarefas. Algumas dessas tarefas podem ser começadas apenas depois que outras tarefas estejam terminadas. Uma ordem parcial nas tarefas é montada considerando tarefa $X \prec$ tarefa Y se a tarefa Y não puder começar até que a tarefa X tenha sido completada. O diagrama de Hasse para as sete tarefas, relativo a esta ordem parcial, é mostrado na Figura 10. Encontre uma ordem na qual essas tarefas possam ser realizadas para completar o projeto.

Solução: Uma ordem das sete tarefas pode ser obtida fazendo uma ordenação topológica. Os passos para uma ordem estão ilustrados na Figura 11. O resultado dessa ordem, $A \prec C \prec B \prec E \prec F \prec D \prec G$, dá uma possível ordem para as tarefas. ◀

Exercícios

1. Quais destas relações em $\{0, 1, 2, 3\}$ são ordenações parciais? Determine as propriedades de uma ordem parcial que estão faltando nas outras.
- **a)** $\{(0, 0), (1, 1), (2, 2), (3, 3)\}$
- **b)** $\{(0, 0) (1, 1), (2, 0), (2, 2), (2, 3), (3, 2), (3, 3)\}$
- **c)** $\{(0, 0), (1, 1), (1, 2), (2, 2), (3, 3)\}$
- **d)** $\{(0, 0), (1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}$
- **e)** $\{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 2), (3, 3)\}$

2. Quais destas relações em $\{0, 1, 2, 3\}$ são ordenações parciais? Determine as propriedades de uma ordem parcial que estão faltando nas outras.
- **a)** $\{(0, 0), (2, 2), (3, 3)\}$
- **b)** $\{(0, 0), (1, 1), (2, 0), (2, 2), (2, 3), (3, 3)\}$
- **c)** $\{(0, 0), (1, 1), (1, 2), (2, 2), (3, 1), (3, 3)\}$
- **d)** $\{(0, 0), (1, 1), (1, 2), (1, 3), (2, 0), (2, 2), (2, 3), (3, 0), (3, 3)\}$
- **e)** $\{(0, 0), (0, 1), (0, 2), (0, 3), (1, 0), (1, 1), (1, 2), (1, 3), (2, 0), (2, 2), (3, 3)\}$

3. (S, R) é um poset se S for o conjunto de todas as pessoas no mundo e $(a, b) \in R$, em que a e b são pessoas, se
- **a)** a é mais alto que b?
- **b)** a não é mais alto que b?
- **c)** $a = b$ ou a é um ancestral de b?
- **d)** a e b têm um amigo em comum?

4. (S, R) é um poset se S for o conjunto de todas as pessoas no mundo e $(a, b) \in R$, em que a e b são pessoas, se
- **a)** a não é mais baixo que b?
- **b)** a pesa mais que b?
- **c)** $a = b$ ou a é um descendente de b?
- **d)** a e b não têm um amigo em comum?

5. Quais destes são posets?
a) $(\mathbf{Z}, =)$ **b)** $(\mathbf{Z}, \neq)$ **c)** $(\mathbf{Z}, \geq)$ **d)** $(\mathbf{Z}, \not|)$

6. Quais destes são posets?
a) $(\mathbf{R}, =)$ **b)** $(\mathbf{R}, <)$ **c)** $(\mathbf{R}, \leq)$ **d)** $(\mathbf{R}, \neq)$

7. Determine se as relações representadas por estas matrizes zero-um são ordens parciais.

a) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$

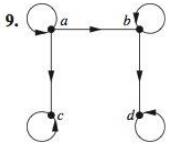
8. Determine se as relações representadas por estas matrizes zero-um são ordens parciais.

a) $\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

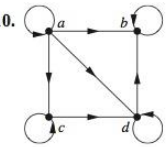
c) $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$

Nos exercícios 9 a 11, determine se a relação com o grafo orientado mostrado é uma ordem parcial.

9.



10.



11.

12. Seja (S, R) um poset. Mostre que (S, R^{-1}) também é um poset, em que R^{-1} é a inversa de R. O poset (S, R^{-1}) é chamado de **dual** de (S, R).

13. Encontre os duais destes posets.

a) $(\{0, 1, 2\}, \leq)$ **b)** $(\mathbf{Z}, \geq)$

c) $(P(\mathbf{Z}), \supseteq)$ **d)** $(\mathbf{Z}^+, |)$

14. Quais destes pares de elementos são comparáveis no poset $(\mathbf{Z}^+, |)$?

a) 5, 15 **b)** 6, 9 **c)** 8, 16 **d)** 7, 7

15. Encontre dois elementos incomparáveis nestes posets.

a) $(P(\{0, 1, 2\}), \subseteq)$ **b)** $(\{1, 2, 4, 6, 8\}, |)$

16. Seja $S = \{1, 2, 3, 4\}$. Com relação à ordem lexicográfica que se baseia na relação usual "menor que",

a) encontre todos os pares em $S \times S$ menores que (2, 3).

b) encontre todos os pares em $S \times S$ maiores que (3, 1).

c) desenhe o diagrama de Hasse do poset $(S \times S, \preccurlyeq)$.

17. Encontre a ordem lexicográfica destas n-uplas:

a) (1, 1, 2), (1, 2, 1) **b)** (0, 1, 2, 3), (0, 1, 3, 2)

c) (1, 0, 1, 0, 1), (0, 1, 1, 1, 0)

18. Encontre a ordem lexicográfica destas seqüências de letras minúsculas da língua inglesa:

a) *quack*, *quick*, *quicksilver*, *quicksand*, *quacking*

b) *open*, *opener*, *opera*, *operand*, *opened*

c) *zoo*, *zero*, *zoom*, *zoology*, *zoological*

19. Encontre a ordem lexicográfica da seqüência de bits 0, 01, 11, 001, 010, 011, 0001 e 0101 com base na ordem $0 < 1$.

20. Desenhe o diagrama de Hasse para a relação "maior que ou igual a" em $\{0, 1, 2, 3, 4, 5\}$.

21. Desenhe o diagrama de Hasse para a relação "menor que ou igual a" em $\{0, 2, 5, 10, 11, 15\}$.

22. Desenhe o diagrama de Hasse para a divisibilidade no conjunto

a) $\{1, 2, 3, 4, 5, 6\}$. **b)** $\{3, 5, 7, 11, 13, 16, 17\}$.

b) $\{2, 3, 5, 10, 11, 15, 25\}$. **d)** $\{1, 3, 9, 27, 81, 243\}$.

23. Desenhe o diagrama de Hasse para a divisibilidade no conjunto

a) $\{1, 2, 3, 4, 5, 6, 7, 8\}$. **b)** $\{1, 2, 3, 5, 7, 11, 13\}$.

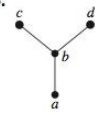
c) $\{1, 2, 3, 6, 12, 24, 36, 48\}$.

d) $(1, 2, 4, 8, 16, 32, 64\}$.

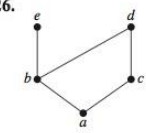
24. Desenhe o diagrama de Hasse para a inclusão no conjunto $P(S)$, em que $S = \{a, b, c, d\}$.

Nos exercícios 25 a 27, liste todos os pares ordenados na ordem parcial com o diagrama de Hasse que acompanha.

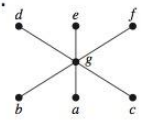
25.



26.



27.



Seja $(S, \preccurlyeq)$ um poset. Dizemos que um elemento $y \in S$ **recobre** um elemento $x \in S$ se $x \prec y$ e não existir nenhum elemento $z \in S$ tal que $x \prec z \prec y$. O conjunto de pares (x, y) tal que y recobre x é chamado de **relação de recobrimento** de $(S, \preccurlyeq)$.

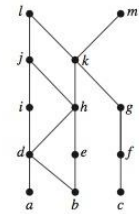
28. Qual é a relação de recobrimento da ordem parcial $\{(a, b) \mid a$ divide $b\}$ em $\{1, 2, 3, 4, 6, 12\}$?

29. Qual é a relação de recobrimento da ordem parcial $\{(A, B) \mid A \subseteq B\}$ no conjunto das partes de S, em que $S = \{a, b, c\}$?

30. Mostre que o par (x, y) pertence à relação de recobrimento de um poset finito $(S, \preccurlyeq)$ se e somente se x for menor do que y e existir uma aresta que liga x e y no diagrama de Hasse desse poset.

31. Mostre que um poset finito pode ser reconstruído a partir de sua relação de recobrimento. [*Dica*: Mostre que o poset é o fecho transitivo reflexivo de sua relação de recobrimento.]

32. Responda a estas questões para a ordem parcial representada por este diagrama de Hasse.



a) Encontre os elementos maximais.
b) Encontre os elementos minimais.
c) Existe um maior elemento?
d) Existe um menor elemento?
e) Encontre todos os limitantes superiores de $\{a, b, c\}$.
f) Encontre o menor limitante superior de $\{a, b, c\}$, se ele existir.
g) Encontre todos os limitantes inferiores de $\{f, g, h\}$.
h) Encontre o maior limitante inferior de $\{f, g, h\}$, se ele existir.

33. Responda a estas questões para o poset $(\{3, 5, 9, 15, 24, 45\}, |)$.
a) Encontre os elementos maximais.
b) Encontre os elementos minimais.
c) Existe um maior elemento?
d) Existe um menor elemento?
e) Encontre todos os limitantes superiores de $\{3, 5\}$.
f) Encontre o menor limitante superior de $\{3, 5\}$, se ele existir.
g) Encontre todos os limitantes inferiores de $\{15, 45\}$.
h) Encontre o maior limitante inferior de $\{15, 45\}$, se ele existir.

34. Responda a estas questões para o poset $(\{2, 4, 6, 9, 12, 18, 27, 36, 48, 60, 72\}, |)$.
a) Encontre os elementos maximais.
b) Encontre os elementos minimais.
c) Existe um maior elemento?
d) Existe um menor elemento?
e) Encontre todos os limitantes superiores de $\{2, 9\}$.
f) Encontre o menor limitante superior de $\{2, 9\}$, se ele existir.
g) Encontre todos os limitantes inferiores de $\{60, 72\}$.
h) Encontre o maior limitante inferior de $\{60, 72\}$, se ele existir.

35. Responda a estas questões para o poset $(\{\{1\}, \{2\}, \{4\}, \{1, 2\}, \{1, 4\}, \{2, 4\}, \{3, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}, \subseteq)$.
a) Encontre os elementos maximais.
b) Encontre os elementos minimais.
c) Existe um maior elemento?
d) Existe um menor elemento?
e) Encontre todos os limitantes superiores de $\{\{2\}, \{4\}\}$.
f) Encontre o menor limitante superior de $\{\{2\}, \{4\}\}$, se ele existir.
g) Encontre todos os limitantes inferiores de $\{\{1, 3, 4\}, \{2, 3, 4\}\}$.
h) Encontre o maior limitante inferior de $\{\{1, 3, 4\}, \{2, 3, 4\}\}$, se ele existir.

36. Dê um poset que tenha
a) um elemento minimal, mas nenhum elemento maximal.
b) um elemento maximal, mas nenhum elemento minimal.
c) nem um elemento maximal nem um elemento minimal.

37. Mostre que a ordem lexicográfica é uma ordem parcial no produto cartesiano de dois posets.

38. Mostre que a ordem lexicográfica é uma ordem parcial no conjunto das seqüências de um poset.

39. Suponha que $(S, \preccurlyeq_1)$ e $(T, \preccurlyeq_2)$ sejam posets. Mostre que $(S \times T, \preccurlyeq)$ é um poset, em que $(s, t) \preccurlyeq (u, v)$ se e somente se $s \preccurlyeq_1 u$ e $t \preccurlyeq_2 v$.

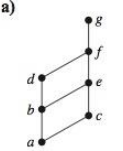
40. **a)** Mostre que existe exatamente um maior elemento de um poset, se esse elemento existir.
b) Mostre que existe exatamente um menor elemento de um poset, se esse elemento existir.

41. **a)** Mostre que existe exatamente um elemento maximal em um poset com um maior elemento.
b) Mostre que existe exatamente um elemento minimal em um poset com um menor elemento.

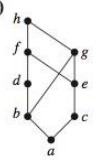
42. **a)** Mostre que o menor limitante superior de um conjunto em um poset é único, se ele existir.
b) Mostre que o maior limitante inferior de um conjunto em um poset é único, se ele existir.

43. Determine se os posets com estes diagramas de Hasse são reticulados.

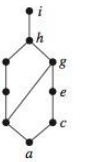
a)



b)



c)



44. Determine se estes posets são reticulados.
a) $(\{1, 3, 6, 9, 12\}, |)$ **b)** $(\{1, 5, 25, 125\}, |)$
c) $(\mathbf{Z}, \geq)$
d) $(P(S), \supseteq)$, em que $P(S)$ é o conjunto das partes de um conjunto S

45. Mostre que todo subconjunto não vazio finito de um reticulado tem um menor limitante superior e um maior limitante inferior.

46. Mostre que se um poset (S, R) for um reticulado, então o poset dual (S, R^{-1}) também será um reticulado.

47. Em uma companhia, o modelo reticulado do fluxo de informações é usado para controlar informações delicadas com classes de segurança representadas por pares ordenados (A, C). Aqui, A é um nível de autoridade, que pode ser não-proprietário (0), proprietário (1), restrito (2) ou registrado (3). Uma categoria C é um subconjunto do conjunto de todos os projetos $\{Cheetah, Impala, Puma\}$. (Nomes de animais são usados com freqüência como nomes em código de projetos em companhias.)
a) É permitido que flua informação de $(Proprietário, \{Cheetah, Puma\})$ para $(Restrito, \{Puma\})$?
b) É permitido que flua informação de $(Restrito, \{Cheetah\})$ para $(Registrado, \{Cheetah, Impala\})$?
c) Para quais classes é permitido que fluam informações de $(Proprietário, \{Cheetah, Puma\})$?
d) De quais classes é permitido que fluam informações para a classe de segurança $(Restrito, \{Impala, Puma\})$?

48. Mostre que o conjunto S das classes de segurança (A, C) é um reticulado, em que A é um inteiro positivo que representa uma classe de autoridade e C é um subconjunto de um conjunto finito de compartimentos, com $(A_1, C_1) \preccurlyeq (A_2, C_2)$ se e somente se $A_1 \leq A_2$ e $C_1 \subseteq C_2$. [*Dica*: Mostre primeiro que $(S, \preccurlyeq)$ é um poset e, então, mostre que o menor limitante superior e o maior limitante inferior de (A_1, C_1) e (A_2, C_2) são $(\max(A_1, A_2), C_1 \cup C_2)$ e $(\min(A_1, A_2), C_1 \cap C_2)$, respectivamente.]

***49.** Mostre que o conjunto de todas as partições de um conjunto S com a relação $P_1 \preccurlyeq P_2$ se a partição P_1 for um refinamento da partição P_2 é um reticulado. (Veja o preâmbulo do Exercício 49 da Seção 8.5.)

50. Mostre que todo conjunto totalmente ordenado é um reticulado.

51. Mostre que todo reticulado finito tem um menor elemento e um maior elemento.

52. Dê um exemplo de um reticulado infinito com
- **a)** nem um maior nem um menor elemento.
- **b)** um menor, mas não um maior elemento.
- **c)** um maior, mas não um menor elemento.
- **d)** tanto um maior quanto um menor elemento.

53. Verifique que $(\mathbf{Z}^+ \times \mathbf{Z}^+, \preccurlyeq)$ é um conjunto bem-ordenado, em que $\preccurlyeq$ é a ordem lexicográfica, como afirmado no Exemplo 8.

54. Determine se cada um destes posets é bem-ordenado.
- **a)** $(S, \leq)$ em que $S = \{10, 11, 12, \ldots\}$
- **b)** $(\mathbf{Q} \cap [0, 1], \leq)$ (o conjunto dos números racionais entre 0 e 1, inclusive)
- **c)** $(S, \leq)$ em que S é o conjunto dos números racionais positivos com denominadores que não excedam 3
- **d)** $(\mathbf{Z}^-, \geq)$ em que $\mathbf{Z}^-$ é o conjunto dos inteiros negativos

Um poset $(R, \preccurlyeq)$ é **bem fundado** se não existirem seqüências decrescentes infinitas de elementos no poset, ou seja, elementos $x_1, x_2, \ldots, x_n$ tal que $\ldots \prec x_n \prec \ldots \prec x_2 \prec x_1$. Um poset $(R, \preccurlyeq)$ é **denso** se para todo $x \in S$ e $y \in S$ com $x \prec y$, existir um elemento $z \in R$ tal que $x \prec z \prec y$.

55. Mostre que o poset $(\mathbf{Z}, \preccurlyeq)$, em que $x \prec y$ se e somente se $|x| < |y|$ é bem fundado, mas não é um conjunto totalmente ordenado.

56. Mostre que um poset denso com, pelo menos, dois elementos que sejam comparáveis não é bem fundado.

57. Mostre que o poset dos números racionais com a relação "menor que ou igual a" usual, $(\mathbf{Q} \leq)$, é um poset denso.

***58.** Mostre que o conjunto das seqüências de letras minúsculas da língua inglesa com a ordem lexicográfica não é nem bem fundado nem denso.

59. Mostre que um poset é bem-ordenado se e somente se ele é totalmente ordenado e bem fundado.

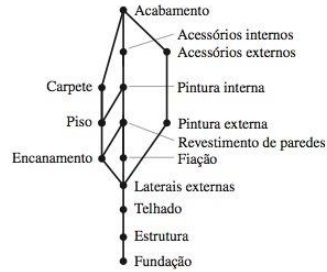
60. Mostre que um poset não vazio finito tem um elemento maximal.

61. Encontre uma ordem total compatível para o poset com o diagrama de Hasse mostrado no Exercício 32.

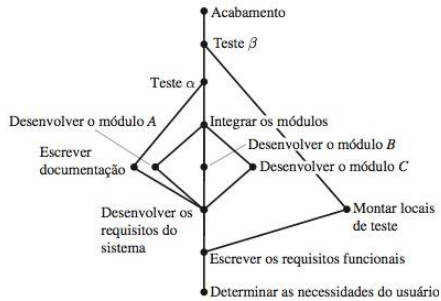
62. Encontre uma ordem total compatível para a relação de divisibilidade no conjunto $\{1, 2, 3, 6, 8, 12, 24, 36\}$.

63. Encontre uma ordem diferente da construída no Exemplo 27 para completar as tarefas no projeto de desenvolvimento.

64. Planeje as tarefas necessárias para construir uma casa, especificando sua ordem, se o diagrama de Hasse que representa essas tarefas é como mostra a figura.



65. Encontre uma ordem de tarefas de um projeto de software se o diagrama de Hasse para as tarefas do projeto é como mostrado.



Termos-chave e Resultados

TERMOS

relações binárias de A em B: um subconjunto de $A \times B$

relação em A: uma relação binária de A em si mesmo (isto é, um subconjunto de $A \times A$)

$S \circ R$: composta de R e S

R^{-1}: relação inversa de R

R^n: potência enésima de R

reflexiva: uma relação R em A é reflexiva se $(a, a) \in R$ para todo $a \in A$

simétrica: uma relação R em A é simétrica se $(b, a) \in R$ sempre que $(a, b) \in R$

anti-simétrica: uma relação de R em A é anti-simétrica se $a = b$ sempre que $(a, b) \in R$ e $(b, a) \in R$

transitiva: uma relação R em A é transitiva se $(a, b) \in R$ e $(b, c) \in R$ implicar que $(a, c) \in R$

relação n-ária em $A_1, A_2, \ldots, A_n$: um subconjunto de $A_1 \times A_2 \times \ldots \times A_n$

modelo relacional de dados: um modelo para representar banco de dados que usa relações n-árias

chave primária: um domínio de uma relação n-ária tal que uma n-upla é determinada de modo único por seu valor nesse domínio

chave composta: o produto cartesiano de domínios de uma relação n-ária tal que uma n-upla é determinada de modo único por seus valores nesses domínios

operador de seleção: uma função que seleciona as n-uplas em uma relação n-ária que satisfazem uma condição especificada

projeção: uma função que produz relações de grau menor a partir de uma relação n-ária deletando campos

junção ou conjunção: uma função que combina relações n-árias que coincidem em determinados campos

grafo orientado ou dígrafo: um conjunto de elementos chamados de vértices e de pares ordenados desses elementos, chamados de arestas

laço: uma aresta da forma (a, a)

fecho de uma relação *R* relativo à propriedade P: a relação S (se existir) que contém R, tem a propriedade **P** e está contida em qualquer relação que contenha R e tenha a propriedade **P**

caminho em um dígrafo: uma seqüência de arestas (a, x_1), (x_1, x_2), …, (x_{n-2}, x_{n-1}), (x_{n-1}, b) tal que o vértice final de cada aresta é o vértice inicial da aresta seguinte na seqüência

circuito (ou ciclo) em um dígrafo: um caminho que começa e termina no mesmo vértice

R* **(relação de conectividade):** a relação que consiste naqueles pares ordenados (a, b) tal que existe um caminho de a a b

relação de equivalência: uma relação reflexiva, simétrica e transitiva

equivalência: se R é uma relação de equivalência, a é equivalente a b se $a\ R\ b$

$[a]_R$ **(classe de equivalência de *a* relativa a *R*)**: o conjunto de todos os elementos de A que são equivalentes a a

$[a]_m$ **(classe de congruência módulo *m*)**: o conjunto dos inteiros congruentes a a módulo m

partição de um conjunto *S*: uma coleção de subconjuntos não vazios dois a dois disjuntos que têm S como união

ordem parcial: uma relação que é reflexiva, anti-simétrica e transitiva

poset (*S*, *R*): um conjunto S e uma ordem parcial R neste conjunto

comparáveis: os elementos a e b no poset $(A, \preccurlyeq)$ são comparáveis se $a \preccurlyeq b$ ou $b \preccurlyeq a$

incomparáveis: elementos em um poset que não são comparáveis

ordem total (ou linear): uma ordem parcial para a qual todo par de elementos é comparável

conjunto totalmente (ou linearmente) ordenado: um poset com uma ordem total (ou linear)

conjunto bem-ordenado: um poset $(S, \preccurlyeq)$, em que $\preccurlyeq$ é uma ordem total e todo subconjunto não vazio de S tem um menor elemento

ordem lexicográfica: uma ordem parcial de produtos cartesianos ou seqüências (ver páginas 569 a 570)

diagrama de Hasse: uma representação gráfica de um poset na qual laços e todas as arestas resultantes da propriedade transitiva não são mostrados, e o sentido das arestas é indicado pela posição dos vértices

elemento maximal: um elemento de um poset que não é menor que nenhum outro elemento do poset

elemento minimal: um elemento de um poset que não é maior que nenhum outro elemento do poset

maior elemento: um elemento de um poset que é maior que ou igual a todos os outros elementos desse conjunto

menor elemento: um elemento de um poset que é menor que ou igual a todos os outros elementos desse conjunto

limitante superior de um conjunto: um elemento em um poset que é maior que todos os elementos desse conjunto

limitante inferior de um conjunto: um elemento em um poset que é menor que todos os outros elementos desse conjunto

menor limitante superior de um conjunto: um limitante superior do conjunto que é menor que todos os outros limitantes superiores

maior limitante inferior de um conjunto: um limitante inferior do conjunto que é maior que todos os outros limitantes inferiores

reticulado: um conjunto parcialmente ordenado, no qual quaisquer dois elementos têm um maior limitante inferior e um menor limitante superior

ordem total compatível com uma ordem parcial: uma ordem total que contém uma ordem parcial dada

ordenação topológica: a construção de uma ordem total compatível com uma ordem parcial dada

RESULTADOS

O fecho reflexivo de uma relação R em um conjunto A é igual a $R \cup \Delta$, em que $\Delta = \{(a, a) \mid a \in A\}$.

O fecho simétrico de uma relação R em um conjunto A é igual a $R \cup R^{-1}$, em que $R^{-1} = \{(b, a) \mid (a, b) \in R\}$.

O fecho transitivo de um relação é igual à relação de conectividade formada a partir desta relação.

O algoritmo de Warshall para encontrar o fecho transitivo de uma relação (ver páginas 550 a 553).

Seja R uma relação de equivalência. Então, as três afirmações a seguir são equivalentes: (1) $a\ R\ b$; (2) $[a]_R \cap [b]_R \neq \emptyset$; (3) $[a]_R = [b]_R$.

As classes de equivalência de uma relação de equivalência em um conjunto A formam uma partição de A. Reciprocamente, é possível construir uma relação de equivalência a partir de qualquer partição, de modo que as classes de equivalência sejam os subconjuntos da partição.

O princípio de indução da boa ordem.

O algoritmo de ordenação topológica (ver páginas 576 a 578).

Questões de Revisão

1. **a)** O que é uma relação em um conjunto?
b) Quantas relações existem em um conjunto com n elementos?

2. **a)** O que é uma relação reflexiva?
b) O que é uma relação simétrica?
c) O que é uma relação anti-simétrica?
d) O que é uma relação transitiva?

3. Dê um exemplo de uma relação no conjunto $\{1, 2, 3, 4\}$ que seja
a) reflexiva, simétrica e não transitiva.
b) não reflexiva, simétrica e transitiva.
c) reflexiva, anti-simétrica e não transitiva.
d) reflexiva, simétrica e transitiva.
e) reflexiva, anti-simétrica e transitiva.

4. a) Quantas relações reflexivas existem em um conjunto com n elementos?
b) Quantas relações simétricas existem em um conjunto com n elementos?
c) Quantas relações anti-simétricas existem em um conjunto com n elementos?

5. a) Explique como uma relação n-ária pode ser usada para representar informação sobre estudantes em uma universidade.
b) Como a relação 5-ária que contém nomes de estudantes, seus endereços, números de telefones, curso e média geral pode ser usada para formar uma relação 3-ária com os nomes dos estudantes, seus cursos e sua média geral?
c) Como a relação 4-ária, que contém nomes de estudantes, seus endereços, números de telefones e cursos, e a relação 4-ária, que contém nomes de estudantes, seus números de identificação, cursos e números de créditos, podem ser combinadas em uma única relação n-ária?

6. a) Explique como usar uma matriz zero-um para representar uma relação em um conjunto finito.
b) Explique como usar uma matriz zero-um que represente uma relação para determinar se a relação é reflexiva, simétrica e/ou anti-simétrica.

7. a) Explique como usar um grafo orientado para representar uma relação em um conjunto finito.
b) Explique como usar um grafo orientado que represente uma relação para determinar se a relação é reflexiva, simétrica e/ou anti-simétrica.

8. a) Defina o fecho reflexivo e o fecho simétrico de uma relação.
b) Como você pode construir o fecho reflexivo de uma relação?
c) Como você pode construir o fecho simétrico de uma relação?
d) Encontre o fecho reflexivo e o fecho simétrico da relação $\{(1, 2), (2, 3), (2, 4), (3, 1)\}$ no conjunto $\{1, 2, 3, 4\}$.

9. a) Defina o fecho transitivo de uma relação.
b) O fecho transitivo de uma relação pode ser obtido incluindo todos os pares (a, c) tal que (a, b) e (b, c) pertençam à relação?
c) Descreva dois algoritmos para encontrar o fecho transitivo de uma relação.
d) Encontre o fecho transitivo da relação $\{(1,1), (1,3), (2,1)$ $(2,3), (2,4), (3,2), (3,4), (4,1)\}$.

10. a) Defina uma relação de equivalência.
b) Quais relações no conjunto $\{a, b, c, d\}$ são relações de equivalência e contêm (a, b) e (b, d)?

11. a) Mostre que a congruência módulo m é uma relação de equivalência sempre que m for um inteiro positivo.
b) Mostre que a relação $\{(a, b) \mid a \equiv \pm b \pmod 7\}$é uma relação de equivalência no conjunto dos inteiros.

12. a) Quais são as classes de equivalência de uma relação de equivalência?
b) Quais são as classes de equivalência da relação de "congruência módulo 5"?
c) Quais são as classes de equivalência da relação de equivalência da Questão 11(b)?

13. Explique a ligação entre relações de equivalência em um conjunto e partições desse conjunto.

14. a) Defina uma ordem parcial.
b) Mostre que a relação de divisibilidade no conjunto dos inteiros positivos é uma ordem parcial.

15. Explique como ordens parciais nos conjuntos A_1 e A_2 podem ser usadas para definir uma ordem parcial no conjunto $A_1 \times A_2$.

16. a) Explique como construir um diagrama de Hasse de uma ordem parcial em um conjunto finito.
b) Trace o diagrama de Hasse da relação de dividibilidade no conjunto $\{2, 3, 5, 9, 12, 15, 18\}$.

17. a) Defina um elemento maximal de um poset e o maior elemento de um poset.
b) Dê um exemplo de um poset que tenha três elementos maximais.
c) Dê um exemplo de um poset com um maior elemento.

18. a) Defina um reticulado.
b) Dê um exemplo de um poset com cinco elementos que seja um reticulado e um exemplo de um poset com cinco elementos que não seja um reticulado.

19. a) Mostre que todo subconjunto finito de um reticulado tem um maior limitante inferior e um menor limitante superior.
b) Mostre que todo reticulado com um número finito de elementos tem um menor elemento e um maior elemento.

20. a) Defina um conjunto bem-ordenado.
b) Descreva um algoritmo para produzir um conjunto bem-ordenado a partir de um conjunto parcialmente ordenado.
c) Explique como o algoritmo de (b) pode ser usado para ordenar tarefas em um projeto se cada tarefa só pode ser feita depois de uma ou mais das outras tarefas estarem concluídas.

Exercícios Complementares

1. Seja S o conjunto de todas as seqüências de letras da língua inglesa. Determine se essas relações são reflexivas, irreflexivas, simétricas, anti-simétricas e/ou transitivas.
a) $R_1 = \{(a, b) \mid a$ e b não têm letras em comum$\}$
b) $R_2 = \{(a, b) \mid a$ e b não têm o mesmo comprimento$\}$
c) $R_3 = \{(a, b) \mid a$ é mais longa que $b\}$

2. Construa uma relação no conjunto $\{a, b, c, d\}$ que seja
a) reflexiva, simétrica, mas não transitiva.
b) irreflexiva, simétrica e transitiva.
c) irreflexiva, anti-simétrica e não transitiva.
d) reflexiva, nem simétrica nem anti-simétrica e transitiva.
e) nem reflexiva, nem irreflexiva, nem simétrica, nem anti-simétrica, nem transitiva.

3. Mostre que a relação R em $\mathbf{Z} \times \mathbf{Z}$ definida por $(a, b)\, R\, (c, d)$ se e somente se $a + d = b + c$ é uma relação de equivalência.
4. Mostre que um subconjunto de uma relação anti-simétrica também é anti-simétrico.
5. Seja R uma relação reflexiva em um conjunto A. Mostre que $R \subseteq R^2$.
6. Suponha que R_1 e R_2 sejam relações reflexivas em um conjunto A. Mostre que $R_1 \oplus R_2$ é irreflexiva.
7. Suponha que R_1 e R_2 sejam relações reflexivas em um conjunto A. $R_1 \cap R_2$ também é reflexiva? $R_1 \cup R_2$ também é reflexiva?
8. Suponha que R seja uma relação simétrica em um conjunto A. $\overline{R}$ também é simétrica?
9. Sejam R_1 e R_2 relações simétricas. $R_1 \cap R_2$ também é simétrica? $R_1 \cup R_2$ também é simétrica?
10. Uma relação R é chamada de **circular** se $a\,R\,b$ e $b\,R\,c$ implicar que $c\,R\,a$. Mostre que R é reflexiva e circular se e somente se ela for uma relação de equivalência.
11. Mostre que uma chave primária em uma relação n-ária é uma chave primária em qualquer projeção dessa relação que contenha essa chave como um de seus campos.
12. Uma chave primária em uma relação n-ária também é uma chave primária em uma relação maior obtida tomando a conjunção desta relação com outra relação?
13. Mostre que o fecho reflexivo do fecho simétrico de uma relação é o mesmo que o fecho simétrico do seu fecho reflexivo.
14. Seja R a relação no conjunto de todos os matemáticos que contém o par ordenado (a, b) se e somente se a e b escreveram um artigo em conjunto.
    a) Descreva a relação R^2.
    b) Descreva a relação R^*.
    c) O **número de Erdös** de um matemático é 1 se este matemático escreveu um artigo com o prolífico matemático húngaro Paul Erdös, é 2 se este matemático não escreveu um artigo em conjunto com Erdös, mas escreveu um artigo em conjunto com alguém que escreveu um artigo em conjunto com Erdös, e assim por diante (exceto que o número de Erdös do próprio Erdös é 0). Dê uma definição do número de Erdös em termos de caminhos em R.
15. **a)** Dê um exemplo para mostrar que o fecho transitivo do fecho simétrico de uma relação não é necessariamente o mesmo que o fecho simétrico do fecho transitivo desta relação.
    b) Mostre, entretanto, que o fecho transitivo do fecho simétrico de uma relação deve conter o fecho simétrico do fecho transitivo desta relação.
16. **a)** Seja S o conjunto das sub-rotinas de um programa de computador. Defina a relação R por $\mathbf{P}\ R\ \mathbf{Q}$ se a sub-rotina **P** chama a sub-rotina **Q** durante sua execução. Descreva o fecho transitivo de R.
    b) Para quais sub-rotinas **P**, (**P**, **P**) pertence ao fecho transitivo de R?
    c) Descreva o fecho reflexivo do fecho transitivo de R.
17. Suponha que R e S sejam relações em um conjunto A com $R \subseteq S$ tal que ambos os fechos de R e S relativos à propriedade **P** existam. Mostre que o fecho de R relativo a **P** é um subconjunto do fecho de S relativo a **P**.
18. Mostre que o fecho simétrico da união de duas relações é a união de seus fechos simétricos.

Links

PAUL ERDŐS (1913–1996) Paul Erdös, nascido em Budapeste, na Hungria, era filho de dois professores de matemática do ensino médio. Ele foi uma criança prodígio; com três anos, sabia multiplicar, de cabeça, números com três algarismos, e com quatro anos descobriu sozinho os números negativos. Como sua mãe não queria expô-lo a doenças contagiosas, ele estudava e tinha aulas apenas em casa. Com 17 anos, Erdös ingressou na Universidade de Eötvös, formando-se quatro anos mais tarde com um doutorado em matemática. Depois de se formar, passou quatro anos em Manchester, na Inglaterra, com uma bolsa de pós-doutorado. Em 1938, ele foi para os Estados Unidos por causa da difícil situação política na Hungria,, especialmente para os judeus. Ele passou boa parte de sua vida nos Estados Unidos, exceto entre 1954 e 1962, quando foi banido como parte da paranóia da era McCarthy. Ele também passou um bom tempo em Israel.

Erdös fez muitas contribuições significativas à combinatória e à teoria dos números. Uma das descobertas das quais ele tinha mais orgulho era de sua demonstração elementar (no sentido de que não usa nenhuma análise complexa) do Teorema dos Números Primos, que fornece uma estimativa para o número de primos que não exceda um inteiro positivo fixo. Ele também participou no desenvolvimento moderno da teoria de Ramsey.

Erdös viajou muito pelo mundo para trabalhar com outros matemáticos, visitando conferências, universidades e laboratórios de pesquisa. Ele não tinha uma morada permanente. Ele se dedicou quase inteiramente à matemática, viajando de um matemático para o outro, proclamando "Meu cérebro está aberto". Erdös foi autor ou co-autor de mais de 1,5 mil artigos e teve mais de 500 co-autores. Cópias de seus artigos são mantidas por Ron Graham, um famoso matemático discreto com quem ele colaborou extensivamente e que cuidou de muitas de suas necessidades mundanas.

Erdös ofereceu recompensas, que iam de $ 10 a $ 10 mil, para a solução de problemas que ele considerava particularmente interessantes, com o tamanho da recompensa dependendo da dificuldade do problema. Ele gastou perto de $ 4 mil. Erdös tinha sua própria linguagem, usando termos como "epsilon" (criança), "chefe" (mulher), "escravo" (homem), "capturado" (casado), "liberado" (divorciado), "Supremo Fascista" (Deus), "Sam" (Estados Unidos) e "Joe" (União Soviética). Embora tivesse curiosidade sobre muitas coisas, concentrou quase toda sua energia em pesquisa matemática. Ele não tinha passatempos nem emprego em tempo integral. Ele nunca se casou e aparentemente permaneceu celibatário. Erdös era extremamente generoso, doando muito do dinheiro que ganhou em prêmios, recompensas e salários para bolsas e causas de valor. Ele viajava com muito pouca bagagem e não gostava de ter muitas possessões materiais.

*19. Descreva um algoritmo, que se baseia no conceito de vértices interiores, que encontre o comprimento do maior caminho entre dois vértices de um grafo orientado, ou determine que existem caminhos arbitrariamente longos entre esses vértices.

20. Quais destas relações são de equivalência no conjunto de todas as pessoas?
   a) $\{(x, y) \mid x$ e y têm o mesmo signo$\}$
   b) $\{(x, y) \mid x$ e y nasceram no mesmo ano$\}$
   c) $\{(x, y) \mid x$ e y estiveram na mesma cidade$\}$

*21. Quantas relações de equivalência diferentes com exatamente três classes de equivalência diferentes existem em um conjunto com cinco elementos?

22. Mostre que $\{(x, y) \mid x - y \in \mathbf{Q}\}$ é uma relação de equivalência no conjunto dos números reais, em que $\mathbf{Q}$ indica o conjunto dos números racionais. O que são $[1]$, $\left[\frac{1}{2}\right]$ e $[\pi]$?

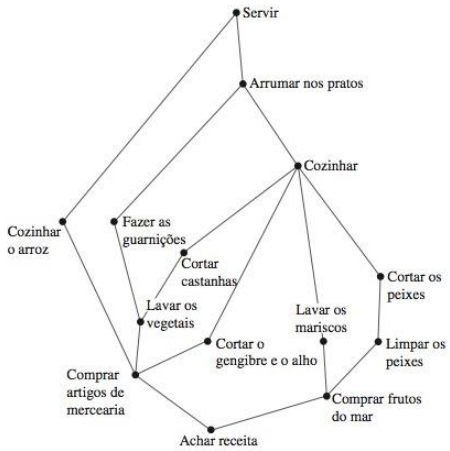
23. Suponha que $P_1 = \{A_1, A_2, \ldots, A_m\}$ e $P_2 = \{B_1, B_2, \ldots, B_n\}$ são ambas partições de um conjunto S. Mostre que a coleção de subconjuntos não vazios da forma $A_i \cap B_j$ é uma partição de S, que é um refinamento tanto de P_1 como de P_2 (veja o preâmbulo do Exercício 49 da Seção 8.5).

*24. Mostre que o fecho transitivo do fecho simétrico do fecho reflexivo de uma relação R é a menor relação de equivalência que contém R.

25. Seja $\mathbf{R}(S)$ o conjunto de todas as relações de um conjunto S. Defina a relação $\preccurlyeq$ em $\mathbf{R}(S)$ por $R_1 \preccurlyeq R_2$ se $R_1 \subseteq R_2$, em que R_1 e R_2 são relações em S. Mostre que $(\mathbf{R}(S), \preccurlyeq)$ é um poset.

26. Seja $\mathbf{P}(S)$ o conjunto de todas as partições de um conjunto S. Defina a relação $\preccurlyeq$ em $\mathbf{P}(S)$ por $P_1 \preccurlyeq P_2$ se P_1 for um refinamento de P_2 (veja o Exercício 49 da Seção 8.5). Mostre que $(\mathbf{P}(S), \preccurlyeq)$ é um poset.

27. Planeje as tarefas necessárias para preparar uma refeição chinesa especificando sua ordem, se o diagrama de Hasse que representa essas tarefas é como mostrado aqui.



Um subconjunto de um poset tal que cada dois elementos deste subconjunto são comparáveis é chamado de **cadeia**. Um subconjunto de um poset é chamado de uma **anticadeia** se cada par de elementos deste subconjunto for incomparável.

28. Encontre todas as cadeias nos posets com os diagramas de Hasse mostrados nos exercícios 25 a 27 da Seção 8.6.

29. Encontre todas as anticadeias nos posets com os diagramas de Hasse mostrados nos exercícios 25 a 27 da Seção 8.6.

30. Encontre uma anticadeia com o maior número de elementos no poset com o diagrama de Hasse do Exercício 32 da Seção 8.6.

31. Mostre que toda cadeia maximal em um poset finito $(S, \preccurlyeq)$ contém um elemento minimal de S. (Uma cadeia maximal é uma cadeia que não é um subconjunto de nenhuma cadeia maior.)

**32. Mostre que todo poset finito pode ser dividido em k cadeias em que k é o maior número de elementos em uma anticadeia nesse poset.

*33. Mostre que em qualquer grupo de $mn + 1$ pessoas existe ou uma lista de $m + 1$ pessoas em que uma pessoa da lista (exceto pela primeira pessoa listada) é um descendente da pessoa anterior da lista ou existem $n + 1$ pessoas tal que nenhuma dessas pessoas é descendente de nenhuma das outras n pessoas. [*Dica*: use o Exercício 32.]

Suponha que $(S, \preccurlyeq)$ seja um conjunto parcialmente ordenado bem fundado. O *princípio da indução bem fundada* diz que $P(x)$ é verdadeira para todo $x \in S$ se $\forall x(\forall y(y \prec x \rightarrow P(y)) \rightarrow P(x))$.

34. Mostre que não é necessário um caso-base separado para o princípio da indução bem fundada. Isto é, $P(u)$ é verdadeira para todos os elementos u minimais de S se $\forall x(\forall y(y \prec x \rightarrow P(y)) \rightarrow P(x))$.

*35. Mostre que o princípio da indução bem fundada é válido.

Uma relação R em um conjunto A é uma **quase-ordem** em A se R for reflexiva e transitiva.

36. Seja R a relação no conjunto de todas as funções de $\mathbf{Z}^+$ para $\mathbf{Z}^+$ tal que (f, g) pertence a R se e somente se f é $O(g)$. Mostre que R é uma quase-ordem.

37. Seja R uma quase-ordem em um conjunto A. Mostre que $R \cap R^{-1}$ é uma relação de equivalência.

*38. Seja R uma quase-ordem e seja S a relação no conjunto das classes de equivalência de $R \cap R^{-1}$ tal que (C, D) pertence a S, em que C e D são classes de equivalência de R se e somente se existirem elementos c de C e d de D tal que (c, d) pertence a R. Mostre que S é uma ordem parcial.

Seja L um reticulado. Defina as operações **encontro** $(\wedge)$ e **conjunção** $(\vee)$ por $x \wedge y = \inf(x, y)$ e $x \vee y = \sup(x, y)$.

39. Mostre que as seguintes propriedades valem para todos os elementos x, y e z de um reticulado L.
   a) $x \wedge y = y \wedge x$ e $x \vee y = y \vee x$ **(propriedades comutativas)**
   b) $(x \wedge y) \wedge z = x \wedge (y \wedge z)$ e $(x \vee y) \vee z = x \vee (y \vee z)$ **(propriedades associativas)**
   c) $x \wedge (x \vee y) = x$ e $x \vee (x \wedge y) = x$ **(propriedades de absorção)**
   d) $x \wedge x = x$ e $x \vee x = x$ **(propriedades idempotentes)**

40. Mostre que se x e y forem elementos de um reticulado L, então $x \vee y = y$ se e somente se $x \wedge y = x$.

Um reticulado L é **limitado** se tiver tanto um **limitante superior**, indicado por 1, tal que $x \preccurlyeq 1$ para todo $x \in L$ quanto um **limitante inferior**, indicado por 0, tal que $0 \preccurlyeq x$ para todo $x \in L$.

41. Mostre que se L for um reticulado limitado com limitante superior 1 e limitante inferior 0, então estas propriedades valem para todos os elementos $x \in L$.

a) $x \vee 1 = 1$ **b)** $x \wedge 1 = x$

c) $x \vee 0 = x$ **d)** $x \wedge 0 = 0$

42. Mostre que todo reticulado finito é limitado.

Um reticulado é denominado **distributivo** se $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ e $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ para todo x, y e z em L.

***43.** Dê um exemplo de um reticulado que não seja distributivo.

44. Mostre que o reticulado $(P(S), \subseteq)$, em que $P(S)$ é o conjunto das partes de um conjunto finito S, é distributivo.

45. O reticulado $(\mathbf{Z}^+, |)$ é distributivo?

O **complemento** de um elemento a de um reticulado limitado L com limitante superior 1 e limitante inferior 0 é um elemento b tal que $a \vee b = 1$ e $a \wedge b = 0$. Esse reticulado é **complementado** se todo elemento do reticulado tiver um complemento.

46. Dê um exemplo de um reticulado finito em que pelo menos um elemento tenha mais que um complemento e pelo menos um elemento não tenha complemento.

47. Mostre que o reticulado $(P(S), \subseteq)$, em que $P(S)$ é o conjunto das partes de um conjunto finito S, é complementado.

***48.** Mostre que se L for um reticulado finito distributivo, então um elemento de L tem, no máximo, um complemento.

O jogo Chomp, introduzido no Exemplo 12 da Seção 1.7, pode ser generalizado para ser jogado em qualquer conjunto finito parcialmente ordenado $(S, \preceq)$ com um menor elemento a. Nesse jogo, um movimento consiste em escolher um elemento x de S e remover x e todos os elementos maiores do que ele de S. Perde o jogador que for forçado a escolher o menor elemento a.

49. Mostre que o jogo Chomp com biscoitos arrumados em uma grade retangular $m \times n$, descrito no Exemplo 12 da Seção 1.7, é o mesmo que o jogo Chomp no poset $(S, |)$, em que S é o conjunto de todos os inteiros posistivos que dividem $p^{m-1}q^{n-1}$, em que p e q são primos distintos.

50. Mostre que se $(S, \preceq)$ tem um maior elemento b, então existe uma estratégia vencedora para o Chomp neste poset. [*Dica*: Generalize o argumento no Exemplo 12 da Seção 1.7.]

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dada uma matriz que representa uma relação em um conjunto finito, determine se a relação é reflexiva e/ou irreflexiva.

2. Dada uma matriz que representa uma relação em um conjunto finito, determine se a relação é simétrica e/ou anti-simétrica.

3. Dada uma matriz que representa uma relação em um conjunto finito, determine se a relação é transitiva.

4. Dado um inteiro positivo n, mostre todas as relações em um conjunto com n elementos.

* **5.** Dado um inteiro positivo n, determine o número de relações transitivas em um conjunto com n elementos.

* **6.** Dado um inteiro positivo n, determine o número de relações de equivalência em um conjunto com n elementos.

* **7.** Dado um inteiro positivo n, mostre todas as relações de equivalência no conjunto dos n menores inteiros positivos.

8. Dada uma relação n-ária, encontre a projeção dessa relação quando campos especificados forem deletados.

9. Dados uma relação m-ária e uma relação n-ária, e um conjunto de campos comuns, encontre a conjunção dessas relações relativa a esses campos comuns.

10. Dada uma matriz que representa uma relação em um conjunto finito, encontre a matriz que representa o fecho reflexivo dessa relação.

11. Dada uma matriz que representa uma relação em um conjunto finito, encontre a matriz que representa o fecho simétrico dessa relação.

12. Dada uma matriz que representa uma relação em um conjunto finito, encontre a matriz que representa o fecho transitivo dessa relação, calculando a conjunção das potências da matriz que representa essa relação.

13. Dada uma matriz que representa uma relação em um conjunto finito, encontre a matriz que representa o fecho transitivo dessa relação, usando o algoritmo de Warshall.

14. Dada uma matriz que representa uma relação em um conjunto finito, encontre a matriz que representa a menor relação de equivalência que contém essa relação.

15. Dada uma ordem parcial em um conjunto finito, encontre uma ordem total compatível com ela, usando ordenação topológica.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Mostre todas as relações diferentes em um conjunto com quatro elementos.
2. Mostre todas as relações reflexivas e simétricas diferentes em um conjunto com seis elementos.
3. Mostre todas as relações transitivas e reflexivas em um conjunto com cinco elementos.
*4. Determine quantas relações transitivas existem em um conjunto com n elementos para todos os inteiros positivos n com $n \leq 7$.
5. Encontre o fecho transitivo de uma relação de sua escolha em um conjunto com, pelo menos, 20 elementos. Use ou uma relação que corresponda a ligações diretas em uma rede de transporte particular ou de comunicações ou use uma relação gerada aleatoriamente.
6. Calcule o número de relações de equivalência diferentes em um conjunto com n elementos para todos os inteiros positivos n que não excedam 20.
7. Mostre todas as relações de equivalência em um conjunto com sete elementos.
*8. Mostre todas as ordens parciais em um conjunto com cinco elementos.
*9. Mostre todos os reticulados em um conjunto com cinco elementos.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Discuta o conceito de uma relação fuzzy. Como as relações fuzzy são usadas?
2. Descreva os princípios básicos dos bancos de dados relacionais, indo além do que foi abrangido na Seção 8.2. Quão amplamente usados são os bancos de dados relacionais quando comparados com outros tipos de bancos de dados?
3. Procure os artigos originais de Warshall e Roy (em francês) nos quais eles desenvolvem algoritmos para encontrar fechos transitivos. Discuta a abordagem deles. Por que você supõe que o que chamamos atualmente de algoritmo de Warshall foi descoberto independentemente por mais de uma pessoa?
4. Descreva como as classes de equivalência podem ser usadas para definir os números racionais como classes de pares de inteiros e como as operações básicas da aritmética em números racionais podem ser definidas seguindo essa abordagem. (Ver Exercício 40 da Seção 8.5.)
5. Explique como Helmut Hasse usou o que agora chamamos de diagramas de Hasse.
6. Descreva alguns dos mecanismos usados para implantar políticas de fluxo de informação em sistemas operados por computador.
7. Discuta o uso do *Program Evaluation and Review Techniche* (PERT) para planejar as tarefas de um grande projeto complicado. Quão amplamente o PERT é usado?
8. Discuta o uso do *Critical Path Method* (CPM) para encontrar o menor tempo para completar um projeto. Quão amplamente o CPM é usado?
9. Discuta o conceito de *dualidade* em um reticulado. Explique como a dualidade pode ser usada para estabelecer novos resultados.
10. Explique o que significa um *reticulado modular*. Descreva algumas das propriedades de reticulados modulares e descreva como os reticulados modulares aparecem no estudo de geometria projetiva.

CAPÍTULO

9 Grafos

9.1 Grafos e Modelos de Grafos
9.2 Terminologia dos Grafos e Tipos Especiais de Grafos
9.3 Representação de Grafos e Isomorfismo de Grafos
9.4 Conectividade
9.5 Caminhos Eulerianos e Hamiltonianos
9.6 Problemas de Caminho mais Curto
9.7 Grafos Planares
9.8 Coloração de Grafos

Os grafos são estruturas discretas que consistem em vértices e arestas que ligam estes vértices. Problemas em quase todas as disciplinas concebíveis podem ser resolvidos usando modelos de grafos. Daremos exemplos para mostrar como eles são usados como modelos em diversas áreas. Por exemplo, mostraremos como são usados para representar a competição de espécies diferentes em um nicho ecológico, para representar quem influencia quem em uma organização e para representar o resultado de torneios. Descreveremos como podem ser usados para modelar convivência entre pessoas, colaboração entre pesquisadores, chamadas telefônicas entre números de telefone e links entre os websites. Mostraremos como podem ser usados para modelar mapas rodoviários e designar tarefas a empregados de uma organização.

Usando modelos de grafos, podemos determinar se é possível percorrer todas as ruas de uma cidade sem passar por uma rua duas vezes e como podemos encontrar o número de cores necessárias para pintar as regiões de um mapa. Os grafos podem ser usados para determinar se um circuito pode ser implementado em uma placa de circuito plano. Distinguiremos dois compostos químicos com a mesma fórmula molecular, mas estruturas diferentes, usando grafos. Podemos determinar se dois computadores estão ligados por um link de comunicação usando modelos de grafo de redes de computadores. Grafos com pesos associados às suas arestas podem ser usados para resolver problemas, tais como encontrar o caminho mais curto entre duas cidades em uma rede de transporte. Podemos também usá-los para planejar exames e associar canais de televisão às estações. Este capítulo introduzirá os conceitos básicos de teorias dos grafos e apresentará muitos modelos de grafos diferentes. Para resolver uma grande variedade de problemas que podem ser estudados usando grafos, introduziremos muitos algoritmos de grafos diferentes. Estudaremos também a complexidade desses algoritmos.

9.1 Grafos e Modelos de Grafos

Começamos com a definição de um grafo.

DEFINIÇÃO 1

Um *grafo* $G = (V, E)$ consiste em V, um conjunto não vazio de *vértices* (ou *nós*), e E, um conjunto de *arestas*. Cada aresta tem um ou dois vértices associados a ela, chamados de suas *extremidades*. Dizemos que uma aresta *liga ou conecta* suas extremidades.

Lembre-se: O conjunto de vértices V de um grafo G pode ser infinito. Um grafo com um conjunto infinito de vértices é chamado de **grafo infinito** e, em comparação, um grafo com um conjuntos finito de vértices é chamado de **grafo finito**. Neste livro, usualmente consideraremos apenas grafos finitos.

Suponha agora que uma rede seja formada de centros de dados e links de comunicação entre computadores. Podemos representar a posição de cada centro de dados por um ponto e cada link de comunicação por um segmento de reta, como mostrado na Figura 1.

Esta rede de computadores pode ser modelada usando um grafo no qual os vértices do grafo representam os centros de dados e as arestas representam os links de comunicação. Em geral, visualizamos os grafos usando pontos para representar os vértices e segmentos de reta, possivelmente curvos, para representar as arestas, em que as extremidades do segmento de reta que representa

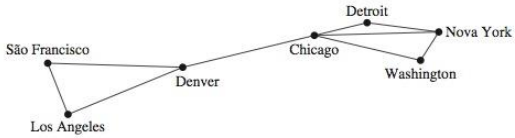


FIGURA 1 Uma Rede de Computadores.

uma aresta são os pontos que representam as extremidades da aresta. Quando desenhamos um grafo, em geral tentamos desenhar as arestas de modo que elas não se cruzem. Entretanto, isso não é necessário porque qualquer descrição que use pontos para representar vértices e qualquer forma de conexão entre os vértices pode ser usada. De fato, existem alguns grafos que não podem ser desenhados no plano sem que as arestas se cruzem (veja a Sessão 9.7). O ponto-chave é que a maneira como desenhamos um grafo é arbitrária, contanto que as conexões corretas entre os vértices sejam mostradas.

Observe que cada aresta do grafo que representa esta rede de computadores conecta dois vértices diferentes, isto é, nenhuma aresta conecta um vértice a si mesmo. Além disso, não existem duas arestas diferentes conectando o mesmo par de vértices. Um grafo no qual cada aresta conecta dois vértices diferentes e duas arestas nunca conectam o mesmo par de vértices é chamado de **grafo simples**. Observe que em um grafo simples, cada aresta está associada a um par não ordenado de vértices e nenhuma outra aresta está associada a esta mesma aresta. Conseqüentemente, quando existe uma aresta de um grafo simples associada a $\{u, v\}$, podemos dizer também, sem possibilidade de confusão, que $\{u, v\}$ é uma aresta do grafo.

Uma rede de computadores pode conter links múltiplos entre centros de dados, como mostrado na Figura 2. Para modelar tais redes, precisamos de grafos que tenham mais de uma aresta conectando o mesmo par de vértices. Os grafos que podem ter **arestas múltiplas** conectando os mesmos vértices são chamados de **multigrafos**. Quando existem m arestas diferentes associadas ao mesmo par não ordenado de vértices $\{u, v\}$, dizemos também que $\{u, v\}$ é uma aresta de multiplicidade m, isto é, podemos pensar neste conjunto de arestas como m cópias diferentes de uma aresta $\{u, v\}$.

Algumas vezes, um link de comunicação conecta um centro de dados com ele mesmo, talvez um laço de auto-alimentação para fins de diagnóstico. Essa rede está ilustrada na Figura 3. Para modelar essa rede, precisamos incluir arestas que conectam um vértice a si mesmo. Essas arestas são chamadas de **laços**. Os grafos que incluem laços, e possivelmente arestas múltiplas que conectam o mesmo par de vértices, são, algumas vezes, chamados de **pseudografos**.

Até agora, os grafos que introduzimos são **grafos não orientados**. Suas arestas também são ditas **não orientadas.** Entretanto, para construir um modelo de grafo, podemos achar necessário atribuir direções às arestas de um grafo. Por exemplo, em uma rede de computadores, alguns links podem operar apenas em um sentido (tais links são chamados de linhas duplex únicas). Isto pode ocorrer se existir uma grande quantidade de tráfico mandada para alguns centros de dados, com pouco ou nenhum tráfico indo em sentido oposto. Essa rede é mostrada na Figura 4.

Para modelar tal rede de computadores, usamos um grafo orientado. Cada aresta de um grafo orientado está associada a um par ordenado. A definição de grafo orientado que damos aqui é mais geral que aquela utilizada no Capítulo 8, em que usamos grafos orientados para representar relações.

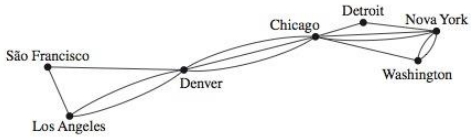


FIGURA 2 Uma Rede de Computadores com Links Múltiplos entre os Centros de Dados.

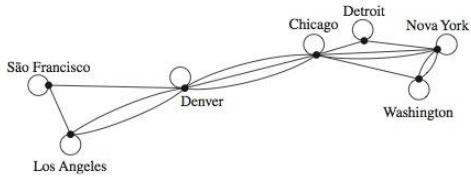


FIGURA 3 **Uma Rede de Computadores com Links de Diagnóstico.**

DEFINIÇÃO 2 Um *grafo orientado* (ou *dígrafo*) (V, E) consiste em um conjunto não vazio de vértices V e um conjunto de *arestas orientadas* (ou *arcos*) E. Cada aresta orientada está associada a um par ordenado de vértices. É dito que aresta orientada associada ao par ordenado (u, v) *começa* em u e *termina* em v.

Quando descrevemos um grafo orientado com um desenho de curvas, usamos uma flecha apontando de u para v para indicar o sentido de uma aresta que começa em u e termina em v. Um grafo orientado pode conter laços e pode conter arestas orientadas múltiplas que começam e terminam nos mesmos vértices. Um grafo orientado pode conter também arestas orientadas que conectam os vértices u e v em ambos os sentidos; isto é, quando um dígrafo contém uma aresta de u para v, ele pode também conter uma ou mais arestas de v para u. Observe que obtemos um grafo orientado quando associamos um sentido a cada aresta de um grafo não orientado. Quando um grafo orientado não tem laços nem arestas orientadas múltiplas, ele é chamado de **grafo orientado simples**. Como um grafo orientado simples tem, no máximo, um aresta associada a cada par ordenado de vértices (u, v), dizemos que (u, v) é uma aresta se existir uma aresta associada a ele no grafo.

Em algumas redes de computadores, links de comunicação múltiplos entre dois centros de dados podem estar presentes, como ilustrado na Figura 5. Os grafos orientados que podem ter **arestas orientadas múltiplas** de um vértice para um segundo vértice (possivelmente o mesmo) são usados para modelar tais redes. Chamamos esses grafos de **multigrafos orientados**. Quando existirem m arestas orientadas, cada uma delas associadas a um par ordenados de vértices (u, v), dizemos que (u, v) é uma aresta de **multiplicidade** m.

Para alguns modelos, podemos precisar de um grafo em que algumas arestas sejam não orientadas, enquanto outras sejam orientadas. Um grafo com tanto arestas orientadas quanto não orientadas é chamado de **grafo misto**. Por exemplo, um grafo misto poderia ser usado para modelar uma rede de computadores que contenha links que operem em ambos os sentidos e outros links que operem em apenas um sentido.

Esta terminologia para os diversos tipos de grafos está resumida na Tabela 1. Algumas vezes usaremos o termo **grafo** como um termo geral para descrever grafos com arestas orientadas ou não orientadas (ou ambas), com ou sem laços, e com ou sem arestas múltiplas. Outras vezes, quando o contexto estiver claro, usaremos o termo grafo para nos referir apenas a grafos não orientados.

Links Por causa do interesse relativamente moderno em teoria dos grafos e pelo fato de ela ter aplicações a uma ampla variedade de disciplinas, muitas terminologias diferentes de teoria dos grafos têm sido introduzidas. O leitor deve determinar como tais termos estão sendo usados sempre

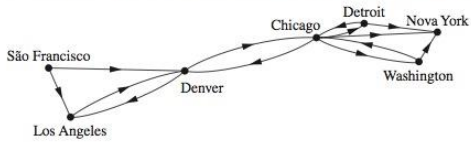


FIGURA 4 **Uma Rede de Comunicações com Links de Comunicações de Mão Única.**

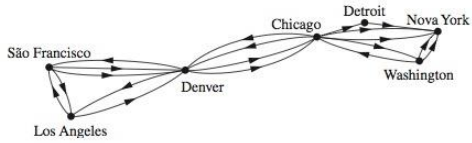


FIGURA 5 **Uma Rede de Computadores com Links Múltiplos de Mão Única.**

que eles forem encontrados. A terminologia usada pelos matemáticos para descrever os grafos tem sido cada vez mais padronizada, mas a terminologia usada para discutir grafos quando eles estão sendo usados em outras disciplinas ainda é bastante variada.

Embora a terminologia usada para descrever grafos possa variar, três questões-chave podem nos ajudar a entender a estrutura de um grafo:

- As arestas de um grafo são não orientadas ou orientadas (ou ambas)?
- Se o grafo for não orientado, estão presentes arestas múltiplas conectando o mesmo par de vértices? Se o grafo for orientado, estão presentes arestas orientadas múltiplas?
- Existem laços presentes?

Responder tais questões nos ajuda a entender os grafos. É menos importante lembrar a terminologia particular que está sendo usada.

Modelos de Grafos

Links

Os grafos são usados em uma ampla variedade de modelos. Aqui, apresentaremos uns poucos modelos de grafos de diversos campos. Outros serão introduzidos nas seções subseqüentes deste e dos capítulos posteriores. Quando construímos um modelo de grafo, precisamos ter certeza de que respondemos corretamente às três perguntas que propusemos sobre a estrutura de um grafo.

EXEMPLO 1

Links

Exemplos Extras

Grafos de Superposição de Nichos em Ecologia Os grafos são usados em muitos modelos que envolvem a interação entre diferentes espécies de animais. Por exemplo, a competição entre espécies em um ecossistema pode ser modelada usando um **grafo de superposição de nicho**. Cada espécie é representada por um vértice. Uma aresta não orientada conecta dois vértices se as duas espécies representadas por esses vértices competem (isto é, algumas das fontes de alimentos que elas usam são as mesmas). Um grafo de superposição de nichos é um grafo simples, pois não são necessários nem laços nem arestas múltiplas neste modelo. O grafo na Figura 6 modela o ecossistema de uma floresta. Vemos a partir deste grafo que esquilos e guaxinins competem, mas que corvos e musaranhos não. ◀

EXEMPLO 2

Links

Grafos de Relacionamento Podemos usar modelos de grafos para representar diversas relações entre as pessoas. Por exemplo, podemos usar um grafo simples para representar se duas pessoas se conhecem, ou seja, se elas têm um relacionamento. Cada pessoa, em um grupo particular

TABELA 1 **Terminologia dos Grafos.**

Tipo	*Arestas*	*Arestas Múltiplas Permitidas?*	*Laços Permitidos?*
Grafo simples	Não orientadas	Não	Não
Multigrafo	Não orientadas	Sim	Não
Pseudografo	Não orientadas	Sim	Sim
Grafo orientado simples	Orientadas	Não	Não
Multigrafo orientado	Orientadas	Sim	Sim
Grafo misto	Orientadas e não orientadas	Sim	Sim

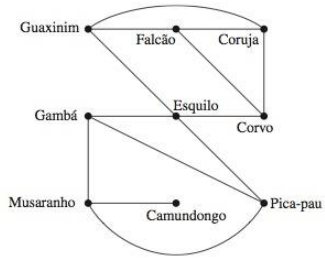


FIGURA 6 Um Grafo de Superposição de Nichos.

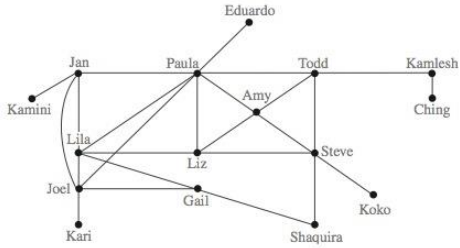


FIGURA 7 Um Grafo de Relacionamento.

de pessoas, é representada por um vértice. Uma aresta não orientada é usada para ligar duas pessoas quando elas se conhecem. Nenhuma aresta múltipla, e em geral nenhum laço, são usados. (Se quiséssemos incluir a noção de autoconhecimento, usaríamos laços.) Um pequeno grafo de relacionamento é mostrado na Figura 7. O grafo de relacionamento de todas as pessoas do mundo tem mais de seis bilhões de vértices e, provavelmente, mais de um trilhão de arestas! Discutiremos mais esse grafo na Seção 9.4. ◀

EXEMPLO 3 **Grafos de Influência** Nos estudos de comportamento de grupo é observado que certas pessoas podem influenciar o pensamento de outras. Um grafo orientado chamado de **grafo de influência** pode ser usado para modelar este comportamento. Cada pessoa do grupo é representada por um vértice. Há uma aresta orientada do vértice a para o vértice b quando a pessoa representada pelo vértice a influencia a pessoa representada pelo vértice b. Este grafo não contém laços nem arestas orientadas múltiplas. Um exemplo de um grafo de influência para membros de um grupo é mostrado na Figura 8. No grupo modelado por este grafo de influência, Deborah pode influenciar Brian, Fred e Linda, mas ninguém pode influenciá-la. Além disso, Yvonne e Brian podem influenciar um ao outro. ◀

EXEMPLO 4 **O Grafo de Hollywood** O **grafo de Hollywood** representa atores por vértices e liga dois vértices quando os atores representados por esses vértices já tiverem trabalhado juntos em um filme. Este grafo é um grafo simples porque suas arestas são não orientadas, ele não contém arestas múltiplas nem laços. De acordo com o Internet Movie Database, em janeiro de 2006 o grafo de Hollywood tinha 637 099 vértices que representavam atores que apareceram em 339 896 filmes, e tinha mais de 20 milhões de arestas. Discutiremos alguns aspectos do grafo de Hollywood mais tarde, na Seção 9.4. ◀

Links

EXEMPLO 5 **Torneios Round-Robin** Um torneio em que cada time joga com cada outro time exatamente uma vez é chamado de **torneio round-robin**. Tais torneios podem ser modelados usando grafos orientados em que cada time é representado por um vértice. Observe que (a, b) é uma aresta se o time a ganhou do time b. Este grafo é um grafo orientado simples, que não contém nem laços nem arestas orientadas múltiplas (pois nem um par de times se enfrenta mais de uma vez). Tal modelo de grafo orientado é apresentado na Figura 9. Vemos que o Time 1 ainda não foi derrotado neste torneio e que o Time 3 ainda não ganhou. ◀

EXEMPLO 6 **Grafos de Colaboração** Um grafo chamado de **grafo de colaboração** pode ser usado para modelar autoria conjunta de artigos acadêmicos. Em um grafo de colaboração, os vértices representam pessoas (talvez restritas a membros de certa comunidade acadêmica) e as arestas ligam

Links

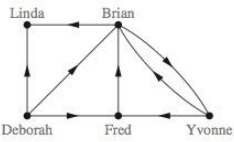


FIGURA 8 Um Grafo de Influência.

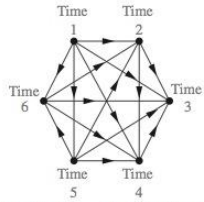


FIGURA 9 Um Modelo de Grafo de um Torneio Round-Robin.

duas pessoas se elas tiverem escrito um artigo em conjunto. Este grafo é um grafo simples porque ele contém arestas não orientadas e não tem laços ou arestas múltiplas. O grafo de colaboração para pessoas que trabalham juntas em artigos de pesquisas em matemática tem mais de 400 000 vértices e 675 000 arestas. Diremos mais sobre este grafo na Seção 9.4. ◀

EXEMPLO 7 **Grafos de Chamadas** Os grafos podem ser usados para modelar chamadas telefônicas feitas em uma rede, tais como uma rede de telefones de longa distância. Em particular, um multigrafo orientado pode ser usado para modelar chamadas, em que cada número de telefone é representado por um vértice e cada chamada telefônica é representada por uma aresta orientada. A aresta que representa uma chamada começa no número de telefone do qual a chamada foi feita e termina no número de telefone para o qual a chamada foi feita. Precisamos de arestas orientadas porque a direção na qual a chamada foi feita importa. Precisamos de arestas orientadas múltiplas porque queremos representar cada chamada feita de um número de telefone particular para um segundo número.

Um pequeno grafo de chamadas telefônicas é mostrado na Figura 10(a), representando sete números de telefones. Esse grafo mostra, por exemplo, que foram feitas três chamadas de 732-555-1234 para 732-555-9876 e duas na direção contrária, mas nenhuma chamada foi feita de 732-555-4444 para qualquer um dos outros seis números, exceto para 732-555-0011. Quando nos importamos apenas se houve uma chamada que liga dois números de telefone, usamos um grafo não orientado com uma aresta que conecta números de telefones quando tiver havido uma chamada entre esses números. Esta versão do grafo de chamadas é mostrada na Figura 10(b).

Grafos de chamadas que modelam atividades telefônicas reais podem ser imensos. Por exemplo, um grafo de chamadas estudado na AT&T, que modela chamadas durante 20 dias, tem cerca de 290 milhões de vértices e 4 bilhões de arestas. Discutiremos mais os grafos de chamadas na Seção 9.4. ◀

EXEMPLO 8 **O Grafo da Web** A World Wide Web pode ser modelada como um grafo orientado no qual cada página da Web é representada por um vértice e no qual uma aresta começa na página *a* da Web e termina na página *b* da Web se existir um link em *a* que direcione para *b*. Como novas páginas da

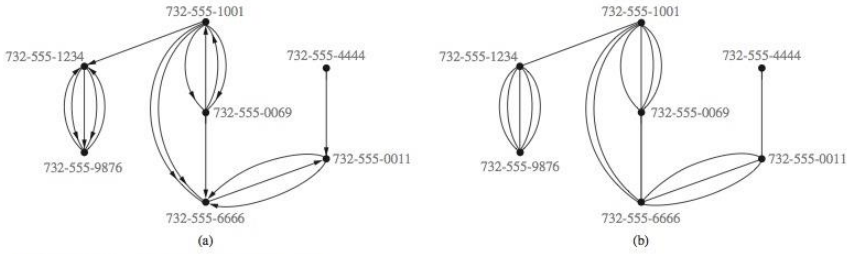


FIGURA 10 Um Grafo de Chamadas.

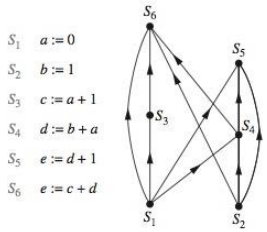


FIGURA 11 **Um Grafo de Precedência.**

Web são criadas e outras removidas em alguma parte da Web a quase todo segundo, o grafo da Web muda numa base quase contínua. Atualmente, o grafo da Web tem mais de três bilhões de vértices e 20 bilhões de arestas. Muitas pessoas estão estudando as propriedades do grafo da Web para entender melhor a natureza da Web. Retornaremos aos grafos da Web na Seção 9.4, e no Capítulo 10 explicaremos como o grafo da Web é usado pelos navegadores da Web que os mecanismos de busca usam para criar índices de páginas da Web. ◀

EXEMPLO 9 **Grafos de Precedência e Processamento Concomitante** Os programas de computador podem ser executados mais rapidamente pela execução de certas instruções concomitantemente. É importante não executar uma instrução que exija resultados de uma instrução que ainda não foi executada. A dependência de instruções das instruções prévias pode ser representada por um grafo orientado. Cada instrução é representada por um vértice e existe uma aresta de um vértice para um segundo vértice se a instrução representada pelo segundo vértice não puder ser executada antes de a instrução representada pelo primeiro vértice ter sido executada. Este grafo é chamado de **grafo de precedência**. Um programa de computador e seu grafo são mostrados na Figura 11. Por exemplo, o grafo mostra que a instrução S_5 não pode ser executada antes das instruções S_1, S_2 e S_4 terem sido executadas. ◀

EXEMPLO 10 **Mapas Rodoviários** Os grafos podem ser usados para modelar mapas rodoviários. Em tais modelos, os vértices representam intersecções e as arestas representam estradas. Arestas não orientadas representam estradas de mão dupla, e as arestas orientadas representam estradas de mão única. As arestas não orientadas múltiplas representam estradas de mão dupla múltiplas que conectam as mesmas duas intersecções. Arestas orientadas múltiplas representam estradas de mão única múltiplas que começam em uma intersecção e terminam na segunda intersecção. Os laços representam estradas em laço. Conseqüentemente, mapas rodoviários que exibem apenas estradas de mão dupla e nenhuma estrada em laço, e nos quais nenhum par de estradas conecta o mesmo par de interseções, podem ser representados usando um grafo não orientado simples. Os mapas rodoviários que exibem apenas estradas de mão única e nenhuma estrada em laço, e nos quais nenhum par de estradas começa na mesma intersecção e termina na mesma intersecção, podem ser modelados usando grafos orientados simples. Grafos mistos são necessários para representar mapas rodoviários que incluam tanto estradas de mão única quanto de mão dupla. ◀

Exercícios

1. Desenhe modelos de grafos, dizendo o tipo do grafo usado (da Tabela 1), para representar rotas de avião nas quais todos os dias existem quatro vôos de Boston para Newark, dois vôos de Newark para Boston, três vôos de Newark para Miami, dois vôos de Miami para Newark, um vôo de Newark para Detroit, dois vôos de Detroit para Newark, três vôos de Newark para Washington, dois vôos de Washington para Newark e um vôo de Washington para Miami, com

a) uma aresta entre vértices que representam cidades que tenham um vôo entre elas (em qualquer sentido).

b) uma aresta entre vértices que representam cidades para cada vôo que opera entre elas (em qualquer sentido).

c) uma aresta entre vértices que representam cidades para cada vôo que opera entre elas (em qualquer sentido), mais um laço para uma viagem especial panorâmica que decola de Miami e aterrisa em Miami.

d) uma aresta de um vértice que representa uma cidade na qual começa um vôo para o vértice que representa a cidade na qual ele termina.

e) uma aresta para cada vôo de um vértice que representa uma cidade onde o vôo começa para o vértice que representa a cidade onde o vôo termina.

2. Que tipo de grafo (da Tabela 1) pode ser usado para modelar um sistema de auto-estradas entre grandes cidades no qual

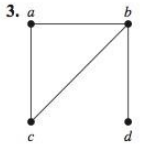
a) existe uma aresta entre os vértices que representam cidades se existir uma auto-estrada interestadual entre elas?

b) existe uma aresta entre os vértices que representam cidades para cada auto-estrada interestadual entre elas?

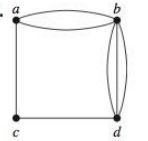
c) existe uma aresta entre os vértices que representam cidades para cada auto-estrada interestadual entre elas e existe um laço no vértice que representa a cidade se existir uma auto-estrada interestadual que circunde esta cidade?

Nos exercícios 3 a 9, determine se o grafo mostrado tem arestas orientadas ou não orientadas, se tem arestas múltiplas, e se tem um ou mais laços. Use suas respostas para determinar o tipo de grafo da Tabela 1 que esse grafo representa.

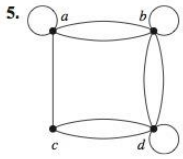
3.



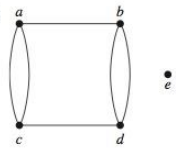
4.



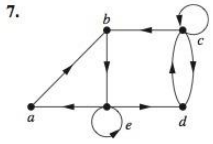
5.



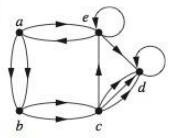
6.



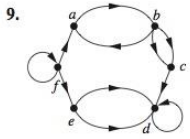
7.



8.



9.



10. Para cada grafo não orientado dos exercícios 3 a 9 que não seja simples, encontre um conjunto de arestas para remover que o torne simples.

11. Seja G um grafo simples. Mostre que a relação R no conjunto de vértices de G, tal que $u\ R\ v$ se e somente se existir uma aresta associada a $\{u, v\}$, é uma relação simétrica irreflexiva em G.

12. Seja G um grafo não orientado com um laço em todo vértice. Mostre que a relação R no conjunto de vértices de G, tal que $u\ R\ v$ se e somente se existir uma aresta associada a $\{u, v\}$, é uma relação simétrica reflexiva em G.

13. O **grafo de intersecção** de uma coleção de conjuntos A_1, A_2, ..., A_n é o grafo que tem um vértice para cada um desses conjuntos e tem uma aresta que liga os vértices que representam dois conjuntos se estes conjuntos tiverem uma intersecção não vazia. Construa o grafo de intersecção destas coleções de conjuntos.

a) $A_1 = \{0, 2, 4, 6, 8\}$, $A_2 = \{0, 1, 2, 3, 4\}$, $A_3 = \{1, 3, 5, 7, 9\}$, $A_4 = \{5, 6, 7, 8, 9\}$, $A_5 = \{0, 1, 8, 9\}$

b) $A_1 = \{..., -4, -3, -2, -1, 0\}$, $A_2 = \{..., -2, -1, 0, 1, 2, ...\}$, $A_3 = \{..., -6, -4, -2, 0, 2, 4, 6, ...\}$, $A_4 = \{..., -5, -3, -1, 1, 3, 5, ...\}$, $A_5 = \{..., -6, -3, 0, 3, 6, ...\}$

c) $A_1 = \{x \mid x < 0\}$, $A_2 = \{x \mid -1 < x < 0\}$, $A_3 = \{x \mid 0 < x < 1\}$, $A_4 = \{x \mid -1 < x < 1\}$, $A_5 = \{x \mid x > -1\}$, $A_6 = \mathbf{R}$

14. Use o grafo de superposição de nichos da Figura 6 para determinar as espécies que competem com os falcões.

15. Construa um grafo de superposição de nichos para 6 espécies de pássaros, em que o *hermit thrush* (espécie de tordo) compete com o *robin* (espécie de tordo) e com o *blue jay* (gaio), o *robin* também compete com o *mockingbird* (pássaro que imita o trinado dos outros), o *mockingbird* também compete com o *blue jay*, e o *nuthatch* (pica-pau cinzento) compete com o *hairy woodpecker* (pica-pau cabeludo).

16. Desenhe um grafo de relacionamento que represente que Tom e Patrícia, Tom e Hope, Tom e Sandy, Tom e Amy, Tom e Marika, Jeff e Patrícia, Jeff e Mary, Patrícia e Hope, Amy e Hope e Amy e Marika se conhecem, mas nenhum dos outros pares de pessoas listadas se conhecem.

17. Podemos usar um grafo para representar se duas pessoas estiveram vivas na mesma época. Desenhe tal grafo para representar se cada par de matemáticos e cientistas da computação com biografias nos primeiros quatro capítulos deste livro e que morreram antes de 1900 foram contemporâneos. (Suponha que duas pessoas viveram na mesma época, se eles estiveram vivos durante o mesmo ano.)

18. Quem pode influenciar Fred e a quem Fred pode influenciar no grafo de influência do Exemplo 3?

19. Construa um grafo de influência para os membros diretores de uma companhia se o Presidente pode influenciar o

Diretor de Pesquisa e Desenvolvimento, o Diretor de Marketing e o Diretor de Operações; o Diretor de Pesquisa e Desenvolvimento pode influenciar o Diretor de Operações; o Diretor de Marketing pode influenciar o Diretor de Operações; e ninguém pode influenciar ou ser influenciado pelo Chefe Financeiro.

20. De que outros times o Time 4 ganhou e de quais times perdeu no torneio representado pelo grafo da Figura 9?

21. Em um torneio, os Tigers ganharam dos Blue Jays, dos Cardinals e dos Orioles, os Blue Jays ganharam dos Cardinals e dos Orioles e os Cardinals ganharam dos Orioles. Modele este resultado com um grafo orientado.

22. Desenhe um grafo de chamadas para os números de telefone 555-0011, 555-1221, 555-1333, 555-8888, 555-2222, 555-0091 e 555-1200, se existiram três chamadas de 555-0011 para 555-8888 e duas chamadas de 555-8888 para 555-0011, duas chamadas de 555-2222 para 555-0091, duas chamadas de 555-1221 para cada um dos outros números e uma chamada de 555-1333 para 555-0011, 555-1221 e 555-1200.

23. Explique como os dois grafos de chamadas telefônicas para as chamadas feitas durante o mês de janeiro e para as chamadas feitas durante o mês de fevereiro podem ser usados para determinar os novos números de telefones das pessoas que mudaram seus números telefônicos.

24. **a)** Explique como os grafos podem ser usados para modelar mensagens de correio eletrônico em uma rede. As arestas devem ser orientadas ou não orientadas? Devem ser permitidas arestas múltiplas? Devem ser permitidos laços?

b) Descreva um grafo que modele as mensagens eletrônicas enviadas em uma rede em uma determinada semana.

25. Como um grafo que modela as mensagens de e-mail enviadas em uma rede pode ser usado para descobrir pessoas que mudaram recentemente seu endereço principal de e-mail?

26. Como um grafo que modela as mensagens de e-mail enviadas em uma rede pode ser usado para descobrir listas de correspondências eletrônicas usadas para enviar a mesma mensagem para muitos endereços diferentes de e-mail?

27. Descreva um modelo de grafo que represente se cada pessoa em uma festa sabe o nome de cada uma das outras pessoas na festa. As arestas devem ser orientadas ou não orientadas? Devem ser permitidas arestas múltiplas? Devem ser permitidos laços?

28. Descreva um modelo de grafo que represente um sistema de metrô em uma grande cidade. As arestas devem ser orientadas ou não orientadas? Devem ser permitidas arestas múltiplas? Devem ser permitidos laços?

29. Descreva um modelo de grafo que represente casamentos tradicionais entre um homem e uma mulher. Este grafo tem alguma propriedade especial?

30. Quais instruções devem ser executadas antes que S_6 seja executada no programa do Exemplo 9? (Use o grafo de precedência da Figura 11.)

31. Construa o grafo de precedência para o seguinte programa:

$S_1: x := 0$
$S_2: x := x + 1$
$S_3: y := 2$
$S_4: z := y$
$S_5: x := x + 2$
$S_6: y := x + z$
$S_7: z := 4$

32. Descreva uma estrutura discreta, que se baseia em um grafo, que possa ser usada para modelar rotas de avião e seus horários de vôo. [*Dica*: Adicione estrutura a um grafo orientado.]

33. Descreva uma estrutura discreta, que se baseia em um grafo, que possa ser usada para modelar relacionamentos entre pares de indivíduos de um grupo, no qual cada indivíduo pode ou gostar ou desgostar ou ser neutro em relação a um outro indivíduo, e a relação inversa pode ser diferente. [*Dica*: Adicione estrutura a um grafo orientado. Trate separadamente as arestas em sentidos opostos entre vértices que representam dois indivíduos.]

34. Descreva um modelo de grafo que possa ser usado para representar todas as formas de comunicação eletrônica entre duas pessoas em único grafo. Que tipo de grafo é necessário?

9.2 Terminologia dos Grafos e Tipos Especiais de Grafos

Introdução

Introduzimos algum vocabulário básico da teoria dos grafos nesta seção. Usaremos este vocabulário mais adiante neste capítulo quando resolvermos muitos tipos diferentes de problemas. Um desses problemas envolve determinar se um grafo pode ser traçado no plano de modo que nenhum par de arestas se intercepte. Um outro exemplo é decidir se existe uma correspondência biunívoca entre os vértices de dois grafos que produza uma correspondência biunívoca entre as arestas dos grafos. Introduziremos também diversas famílias importantes de grafos usados com freqüência como exemplos e em modelos. Serão descritas diversas aplicações importantes nas quais estes tipos de grafos aparecem.

Terminologia Básica

Primeiro, daremos alguma terminologia que descreve os vértices e arestas de grafos não orientados.

DEFINIÇÃO 1 Dois vértices u e v em um grafo não orientado G são ditos *adjacentes* (ou *vizinhos*) em G se u e v são extremidades de uma aresta de G. Se e estiver associado a $\{u, v\}$, a aresta e é dita *incidente aos* vértices u e v. Diz-se também que a aresta e *conecta* u e v. Os vértices u e v são chamados de *extremidades* de uma aresta associada a $\{u, v\}$.

Para descrever quantas arestas são incidentes a um vértice, fazemos a seguinte definição.

DEFINIÇÃO 2 O *grau de um vértice de um grafo não orientado* é o número de arestas incidentes a ele, exceto que um laço em um vértice contribui duas vezes ao grau daquele vértice. O grau do vértice v é indicado por $\text{gr}(v)$.

EXEMPLO 1 Quais são os graus dos vértices nos grafos G e H mostrados na Figura 1?

Solução: Em G, $\text{gr}(a) = 2$, $\text{gr}(b) = \text{gr}(c) = \text{gr}(f) = 4$, $\text{gr}(d) = 1$, $\text{gr}(e) = 3$ e $\text{gr}(g) = 0$. Em H, $\text{gr}(a) = 4$, $\text{gr}(b) = \text{gr}(e) = 6$, $\text{gr}(c) = 1$ e $\text{gr}(d) = 5$. ◀

Um vértice de grau zero é dito **isolado**. É claro que um vértice isolado não é adjacente a nenhum vértice. O vértice g no grafo G do Exemplo 1 é isolado. Um vértice é **pendente** se e somente se ele tem grau 1. Conseqüentemente, um vértice pendente é adjacente a exatamente um outro vértice. O vértice d no grafo G do Exemplo 1 é pendente.

O exame dos graus dos vértices em um modelo de grafo pode fornecer informação útil sobre o modelo, como mostra o Exemplo 2.

EXEMPLO 2 O que representa o grau de um vértice em um grafo de superposição de nichos (introduzidos no Exemplo 1 da Seção 9.1)? Quais vértices neste grafo são pendentes e quais são isolados? Use o grafo de superposição de nichos da Figura 6 da Seção 9.1 para interpretar suas respostas.

Solução: Existe uma aresta entre dois vértices em um grafo de superposição de nichos se e somente se as duas espécies representadas por esses vértices competirem. Logo, o grau de um vértice em um grafo de superposição de nichos é o número de espécies no ecossistema que compete com a espécie representada pelo vértice. Um vértice é pendente se a espécie competir exatamente

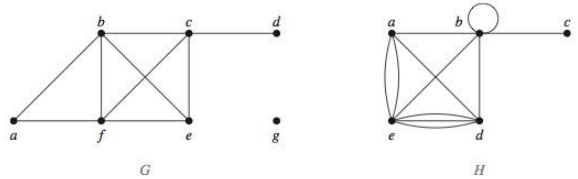


FIGURA 1 **Os Grafos Não Orientados G e H.**

com uma outra espécie no ecossistema. Finamente, o vértice que representa uma espécie é isolado se esta espécie não competir com nenhuma outra espécie no ecossistema.

Por exemplo, o grau do vértice que representa o esquilo na Figura 6 da Seção 9.1 é quatro, porque o esquilo compete com quatro outras espécies: o corvo, o gambá, o guaxinim e o pica-pau. No grafo da Figura 6, o camundongo é a única espécie representada por um vértice pendente, porque ele compete apenas com o musaranho e todas as outras espécies competem com, pelo menos, duas outras espécies. O vértice que representa uma espécie é pendente se esta espécie competir com apenas uma outra espécie. Não existe vértice isolado no grafo da Figura 6, pois todas as espécies neste ecossistema competem com, pelo menos, uma outra espécie. ◀

O que obtemos quando somamos os graus de todos os vértices de um grafo $G = (V, E)$? Cada aresta contribui com dois para a soma dos graus dos vértices, pois uma aresta é incidente a exatamente dois (possivelmente iguais) vértices. Isto significa que a soma dos graus dos vértices é duas vezes o número de arestas. Temos este resultado no Teorema 1, que é algumas vezes chamado de Teorema do Aperto de Mãos, por causa da analogia entre uma aresta ter duas extremidades e um aperto de mãos envolver duas mãos.

TEOREMA 1

TEOREMA DO APERTO DE MÃOS Seja $G = (V, E)$ um grafo não orientado com e arestas. Então

$$2e = \sum_{v \in V} \text{gr}(v).$$

(Observe que isto se aplica mesmo que arestas múltiplas e laços estejam presentes.)

EXEMPLO 3 Quantas arestas existem em um grafo com 10 vértices, cada um de grau 6?

Solução: Como a soma dos graus dos vértices é $6 \cdot 10 = 60$, segue que $2e = 60$. Portanto, $e = 30$. ◀

O Teorema 1 mostra que a soma dos graus dos vértices de um grafo não orientado é par. Este fato simples tem muitas conseqüências, uma das quais é dada no Teorema 2.

TEOREMA 2 Um grafo não orientado tem um número par de vértices de grau ímpar.

Demonstração: Sejam V_1 e V_2 o conjunto de vértices de grau par e o conjunto de vértices de grau ímpar, respectivamente, em um grafo não orientado $G = (V, E)$. Então

$$2e = \sum_{v \in V} \text{gr}(v) = \sum_{v \in V_1} \text{gr}(v) + \sum_{v \in V_2} \text{gr}(v).$$

Como $\text{gr}(v)$ é par para $v \in V_1$, o primeiro termo do lado direito da última igualdade é par. Além disso, a soma dos dois termos do lado direito da última igualdade é par, porque sua soma é $2e$. Logo, o segundo termo na soma também é par. Como todos os termos nesta soma são ímpares, deve existir um número par de tais termos. Logo, existe um número par de vértices de grau ímpar. ◁

A terminologia para grafos com arestas orientadas reflete o fato que arestas em grafos orientados têm direção.

DEFINIÇÃO 3 Quando (u, v) for uma aresta em um grafo G com arestas orientadas, dizemos que u é *adjacente para* v e v é *adjacente a partir de* u. O vértice u é dito *vértice inicial* de (u, v), e v é dito *vértice final* ou *terminal* de (u, v). Os vértices inicial e final de um laço são os mesmos.

Como as arestas em grafos com arestas orientadas são pares ordenados, a definição do grau de um vértice pode ser refinada para refletir o número de arestas que tem este vértice como vértice inicial e como vértice final.

DEFINIÇÃO 4 Em um grafo com arestas orientadas, o *grau de entrada de um vértice* v, indicado por $\text{gr}^-(v)$, é o número de arestas que tem v como seu vértice final. O *grau de saída de* v, indicado por $\text{gr}^+(v)$, é o número de arestas que tem v como seu vértice inicial. (Observe que um laço em um vértice contribui 1 tanto para o grau de entrada quanto para o grau de saída desse vértice.)

EXEMPLO 4 Encontre o grau de entrada e o grau de saída de cada vértice no grafo G com arestas orientadas mostrado na Figura 2.

Solução: Os graus de entrada em G são $\text{gr}^-(a) = 2$, $\text{gr}^-(b) = 2$, $\text{gr}^-(c) = 3$, $\text{gr}^-(d) = 2$, $\text{gr}^-(e) = 3$ e $\text{gr}^-(f) = 0$. Os graus de saída em G são $\text{gr}^+(a) = 4$, $\text{gr}^+(b) = 1$, $\text{gr}^+(c) = 2$, $\text{gr}^+(d) = 2$, $\text{gr}^+(e) = 3$ e $\text{gr}^+(f) = 0$. ◀

Como cada aresta tem um vértice inicial e um vértice final, a soma dos graus de entrada e a soma dos graus de saída de todos os vértices em um grafo com arestas orientadas são as mesmas. Ambas as somas são o número de arestas no grafo. Este resultado está enunciado como Teorema 3.

TEOREMA 3 Seja $G = (V, E)$ um grafo com arestas orientadas. Então

$$\sum_{v \in V} \text{gr}^-(v) = \sum_{v \in V} \text{gr}^+(v) = |E|.$$

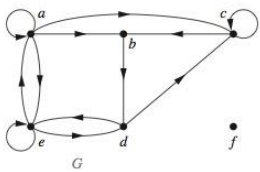


FIGURA 2 O Grafo Orientado *G*.

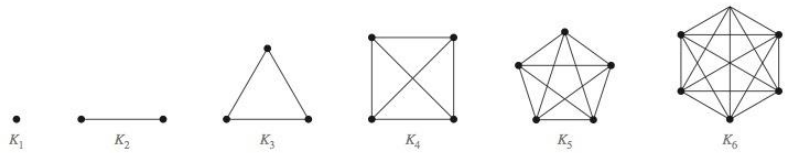


FIGURA 3 **Os Grafos K_n para $1 \leq n \leq 6$.**

Existem muitas propriedades de um grafo com arestas orientadas que não dependem da direção de suas arestas. Conseqüentemente, às vezes é útil ignorar essas direções. O grafo não orientado que resulta de ignorar as direções das arestas é chamado de **grafo não orientado subjacente**. Um grafo com arestas orientadas e seu grafo não orientado subjacente têm o mesmo número de arestas.

Alguns Grafos Simples Especiais

Introduziremos agora diversas classes de grafos simples. Estes grafos são usados freqüentemente como exemplos e aparecem em muitas aplicações.

EXEMPLO 5 **Grafos Completos** O **grafo completo de n vértices**, indicado por K_n, é o grafo simples que contém exatamente uma aresta entre cada par de vértices distintos. Os grafos K_n, para $n = 1, 2, 3, 4, 5, 6$, estão mostrados na Figura 3. ◀

EXEMPLO 6 **Ciclos** O **ciclo C_n, $n \geq 3$**, consiste em n vértices $v_1, v_2, \ldots, v_n$ e arestas $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}$ e $\{v_n, v_1\}$. Os ciclos C_3, C_4, C_5 e C_6 estão mostrados na Figura 4. ◀

EXEMPLO 7 **Rodas** Obtemos a **roda W_n** quando adicionamos mais um vértice ao ciclo C_n, para $n \geq 3$, e conectamos este novo vértice a cada um dos n vértices em C_n, por novas arestas. As rodas W_3, W_4, W_5 e W_6 estão mostradas na Figura 5. ◀

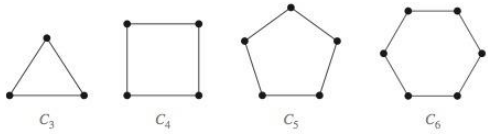


FIGURA 4 **Os Ciclos C_3, C_4, C_5 e C_6.**

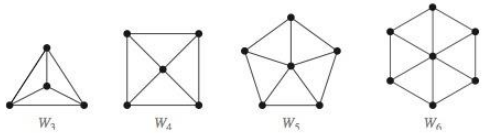


FIGURA 5 **As Rodas W_3, W_4, W_5 e W_6.**

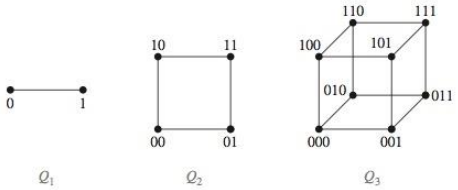


FIGURA 6 **O n-Cubo Q_n para $n = 1, 2$ e 3.**

EXEMPLO 8 ***n*-Cubos** O **hipercubo de dimensão *n*** ou ***n*-cubo**, indicado por Q_n, é o grafo que tem vértices que representam as 2^n seqüências de bit de comprimento n. Dois vértices são adjacentes se e somente se as seqüências de bit que eles representam diferem em exatamente uma posição de bit. Os grafos Q_1, Q_2 e Q_3 estão mostrados na Figura 6. Observe que você pode construir o $(n + 1)$-cubo Q_{n+1} a partir do n-cubo Q_n fazendo duas cópias de Q_n, adicionando um 0 ao início das seqüências de uma cópia de Q_n e um 1 ao início das seqüências da outra cópia de Q_n, e adicionando arestas que conectem dois vértices que tenham seqüências que diferem apenas no primeiro bit. Na Figura 6, Q_3 é construído a partir de Q_2 desenhando duas cópias de Q_2 como as faces superior e inferior de Q_3, adicionando 0 no início da seqüência de cada vértice da face inferior e 1 no início da seqüência de cada vértice da face superior. (Aqui, *face* significa a face de um cubo no espaço tridimensional. Pense no desenho do grafo Q_3 no espaço tridimensional com cópias de Q_2 como as faces superior e inferior de um cubo e depois desenhe a projeção da descrição resultante no plano.) ◀

Grafos Bipartidos

Links

Às vezes, um grafo tem a propriedade que seu conjunto de vértices pode ser dividido em dois subconjuntos disjuntos, tal que cada aresta conecta um vértice de um destes subconjuntos a um vértice do outro subconjunto. Por exemplo, considere o grafo que representa casamentos entre homens e mulheres em um vilarejo, em que cada pessoa é representada por um vértice e um casamento é representado por uma aresta. Neste grafo, cada aresta liga um vértice em um subconjunto de vértices que representa os homens a um vértice em um subconjunto de vértices que representa as mulheres. Isso nos leva à Definição 5.

DEFINIÇÃO 5 Um grafo simples G é dito *bipartido* se o seu conjunto V de vértices pode ser dividido em dois conjuntos disjuntos V_1 e V_2 tal que cada aresta do grafo conecta um vértice em V_1 e um vértice em V_2 (de modo que nenhuma aresta em G conecta dois vértices, seja em V_1, seja em V_2). Quando esta condição é válida, chamamos o par (V_1, V_2) de *bipartição* do conjunto de vértices V de G.

No Exemplo 9, mostraremos que C_6 é bipartido, e no Exemplo 10, mostraremos que K_3 não é bipartido.

EXEMPLO 9 C_6 é bipartido, como mostrado na Figura 7, pois seu conjunto de vértices pode ser dividido em dois conjuntos $V_1 = \{v_1, v_3, v_5\}$ e $V_2 = \{v_2, v_4, v_6\}$, e cada aresta de C_6 conecta um vértice em V_1 e um vértice em V_2. ◀

EXEMPLO 10 K_3 não é bipartido. Para verificar isto, observe que se dividirmos o conjunto de vértices K_3 em dois conjuntos disjuntos, um dos dois conjuntos deve conter dois vértices. Se o grafo fosse bipartido,

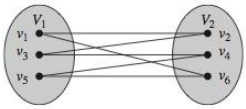


FIGURA 7 **Mostrando que C_6 É Bipartido.**

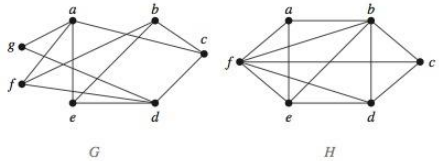


FIGURA 8 **Os Grafos Não Orientados *G* e *H*.**

esses dois vértices não poderiam estar conectados por uma aresta, mas em K_3 cada vértice está conectado a qualquer outro vértice por uma aresta. ◀

EXEMPLO 11 Os grafos G e H mostrados na Figura 8 são bipartidos?

Solução: O grafo G é bipartido porque seu conjunto de vértices é a união de dois conjuntos disjuntos, $\{a, b, d\}$ e $\{c, e, f, g\}$, e cada aresta conecta um vértice em um desses subconjuntos a um vértice no outro subconjunto. (Observe que, para G ser bipartido, não é necessário que todo vértice em $\{a, b, d\}$ seja adjacente a todo vértice em $\{c, e, f, g\}$. Por exemplo, b e g não são adjacentes.)

O grafo H não é bipartido, pois seu conjunto de vértices não pode ser dividido em dois subconjuntos de modo que as arestas não conectem dois vértices do mesmo subconjunto. (O leitor deve verificar isto, considerando os vértices a, b e f.) ◀

O Teorema 4 fornece um critério útil para determinar se um grafo é bipartido.

TEOREMA 4

Um grafo simples é bipartido se e somente se for possível associar uma de duas cores diferentes a cada vértice do grafo de modo que nenhum par de vértices adjacentes tenha a mesma cor associada.

Demonstração: Primeiro, suponha que $G = (V, E)$ seja um grafo simples bipartido. Então, $V = V_1 \cup V_2$, em que V_1 e V_2 são conjuntos disjuntos e toda aresta em E conecta um vértice em V_1 e um vértice em V_2. Se associarmos uma cor a cada vértice em V_1 e uma segunda cor a cada vértice em V_2, então nenhum par de vértices adjacentes tem a mesma cor associada a eles.

Suponha agora que seja possível associar cores aos vértices de um grafo usando apenas duas cores, de modo que nenhum par de vértices adjacentes tenha a mesma cor associada a eles. Seja V_1 o conjunto de vértices associado a uma cor e V_2 o conjunto de vértices associado à outra cor. Então, V_1 e V_2 são disjuntos e $V = V_1 \cup V_2$. Além disso, toda aresta conecta um vértice em V_1 a um vértice em V_2, pois nenhum par de vértices adjacentes está ou em V_1 ou em V_2. Conseqüentemente, G é bipartido. ◁

Ilustramos como o Teorema 4 pode ser usado para determinar se um grafo é bipartido no Exemplo 12.

EXEMPLO 12 Use o Teorema 4 para determinar se os grafos no Exemplo 11 são bipartidos.

Solução: Consideremos primeiro o grafo G. Tentaremos associar uma de duas cores, digamos, vermelho e azul, a cada vértice G, de modo que toda aresta em G conecte um vértice vermelho a um vértice azul. Sem perda de generalidade, começamos associando arbitrariamente vermelho a a. Então, devemos associar azul a c, e, f e g, pois cada um desses vértices é adjacente a a. Para

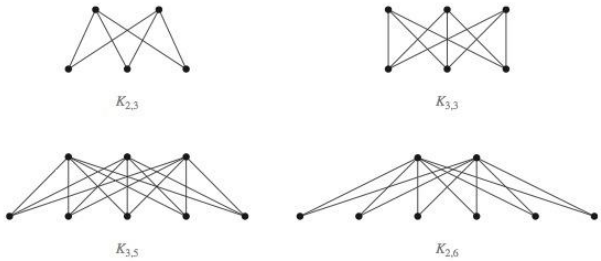


FIGURA 9 Alguns Grafos Bipartidos Completos.

evitar ter uma aresta com duas extremidades azuis, devemos associar vermelho a todos os vértices adjacentes a c, e, f ou g. Isso significa que devemos associar vermelho tanto a b quanto a d (e significa que devemos associar vermelho a a, o que já foi feito). Agora já associamos cores a todos os vértices, com a, b e d vermelhos e c, e, f e g azuis. Verificando todas as arestas, vemos que cada aresta conecta um vértice vermelho a um vértice azul. Portanto, pelo Teorema 4, o grafo G é bipartido.

A seguir, tentaremos associar ou vermelho ou azul a cada vértice em H, de modo que nenhuma aresta em H conecte dois vértices da mesma cor. Sem perda de generalidade, associamos arbitrariamente vermelho a a. Então, devemos associar azul a b, e e f, pois cada um deles é adjacente a a. Mas isto não é possível porque e e f são adjacentes, de modo que eles não podem ambos estarem associados a azul. Este argumento mostra que não podemos associar uma de duas cores a cada um dos vértices de H de modo que não seja associada a mesma cor a nenhum par de vértices adjacentes. Portanto, pelo Teorema 4, o grafo H não é bipartido. ◀

O Teorema 4 é um exemplo de um resultado na parte de teoria dos grafos conhecida como coloração de grafos. A coloração de grafos é uma parte importante da Teoria dos Grafos, com aplicações importantes. Estudaremos mais a coloração de grafos na Seção 9.8.

Um outro critério útil para determinar se um grafo é bipartido é baseado na noção de caminho, um tópico que estudaremos na Seção 9.4. Um grafo é bipartido se e somente se não for possível começar em um vértice e retornar a este vértice percorrendo um número ímpar de arestas distintas. Tornaremos esta noção mais precisa quando discutirmos caminhos e ciclos nos grafos, na Seção 9.4 (veja o Exercício 53 naquela seção).

EXEMPLO 13 **Grafos Bipartidos Completos** O **grafo bipartido completo** $K_{m,n}$ é um grafo que tem seu conjunto de vértices dividido em dois subconjuntos de m e n vértices, respectivamente. Existe uma aresta entre dois vértices se e somente se um vértice estiver no primeiro subconjunto e o outro vértice no segundo subconjunto. Os grafos bipartidos completos $K_{2,3}$, $K_{3,3}$, $K_{3,5}$ e $K_{2,6}$ estão mostrados na Figura 9. ◀

Algumas Aplicações de Tipos Especiais de Grafos

Estudaremos como os grafos bipartidos e os tipos especiais de grafos são usados em modelos nos exemplos 14 a 16.

EXEMPLO 14 **Atribuição de Tarefas** Suponha que existam m empregados em um grupo e j tarefas diferentes que precisam ser feitas, em que $m \leq j$. Cada empregado é treinado para fazer uma ou mais dessas j tarefas. Podemos usar um grafo para modelar as habilidades dos empregados. Representamos cada empregado por um vértice e cada tarefa por um vértice. Para cada empregado, incluímos uma aresta do vértice que representa aquele empregado aos vértices que representam todas as tarefas que o empregado é capaz de fazer.

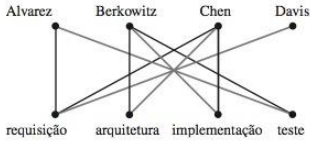


FIGURA 10 Modelando as Tarefas que os Empregados São Capazes de Fazer.

Observe que o conjunto de vértices deste grafo pode ser dividido em dois conjuntos disjuntos, o conjunto dos vértices que representam empregados e o conjunto de vértices que representam tarefas, e cada aresta conecta um vértice que representa um empregado a um vértice que representa uma tarefa. Conseqüentemente, este grafo é bipartido.

Por exemplo, suponha que um grupo tenha quatro empregados: Alvarez, Berkowitz, Chen e Davis; e suponha que seja necessário fazer quatro tarefas para completar um projeto: requisição, arquitetura, implementação e teste. Suponha que Alvarez seja capaz de fazer requisição e teste; Berkowitz seja capaz de fazer arquitetura, implementação e teste; Chen seja capaz de fazer requisição, arquitetura e implementação; e Davis seja capaz de fazer apenas requisição. Podemos modelar estas habilidades dos empregados usando o grafo bipartido mostrado na Figura 10.

Para completar o projeto, devemos designar tarefas aos empregados de modo que toda tarefa tenha um empregado designado a ela e que a nenhum empregado seja atribuída mais de uma tarefa. Neste caso, podemos designar Alvarez para teste, Berkowitz para implementação, Chen para arquitetura e Davis para requisição, como mostrado na Figura 10 (onde as linhas cinzas mostram a atribuição de tarefas). ◀

Encontrar uma atribuição de tarefas para os empregados pode ser pensado como encontrar um emparelhamento num modelo de grafo. Um **emparelhamento** em um grafo simples é um subconjunto do conjunto das arestas do grafo tal que nenhum par de arestas é incidente no mesmo vértice; um **emparelhamento maximal** é um emparelhamento com o maior número possível de arestas. Em outras palavras, um emparelhamento é um subconjunto das arestas tal que se $\{s, t\}$ e $\{u, v\}$ forem arestas do emparelhamento, então s, t, u e v são distintos. Para designar tarefas a empregados, de modo que o maior número possível de empregados tenha tarefas atribuídas a eles, procuramos um emparelhamento máximo no grafo que modela as habilidades dos empregados. (O leitor interessado pode encontrar mais sobre emparelhamentos em livros sobre teoria dos grafos, incluindo [GrYe06].)

EXEMPLO 15 **Redes de Áreas Locais** Os diversos computadores em um prédio, tais como minicomputadores e computadores pessoais, bem como acessórios como impressoras e plotters, podem ser conectados usando uma *rede de área local*. Algumas destas redes são baseadas na *topologia estrela*, em que todos os aparelhos são ligados a um aparelho de controle central. Uma rede de área local pode ser representada usando um grafo bipartido completo $K_{1,n}$, como mostrado na Figura 11(a). As mensagens são enviadas de aparelho a aparelho através do aparelho de controle central.

Links

Outras redes de área local são baseadas na *topologia anel*, na qual cada aparelho é conectado exatamente a dois outros. Redes de área local com uma topologia anel são modeladas usando n-ciclos, C_n, como mostrado na Figura 11(b). As mensagens são enviadas de aparelho em aparelho até que o destinatário da mensagem seja atingido.

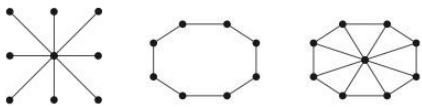


FIGURA 11 Topologias Estrela, Anel e Híbrida para Redes de Área Local.

Finalmente, algumas redes de área local usam um híbrido destas duas topologias. As mensagens podem ser enviadas em torno do anel ou através do aparelho central. Esta redundância torna a rede mais confiável. Redes de área local com esta redundância podem ser modeladas usando rodas W_n, como mostrado na Figura 11(c). ◀

EXEMPLO 16 **Redes de Interconexão para Computação Paralela** Por muitos anos, computadores executaram programas com uma operação de cada vez. Conseqüentemente, os algoritmos escritos para resolver problemas eram projetados para fazer um passo de cada vez; tais algoritmos são chamados de **seriais**. (Quase todos os algoritmos descritos neste livro são seriais.) Entretanto, muitos problemas computacionalmente intensos, como simulações de clima, imagens médicas e criptoanálise, não podem ser resolvidos em uma quantidade razoável de tempo usando operações seriais, mesmo em um supercomputador. Além disso, existe um limite físico para quão rápido um computador pode efetuar as operações básicas, de modo que sempre haverá problemas que não podem ser resolvidos em um intervalo de tempo razoável usando operações seriais.

O **processamento paralelo**, que usa computadores feitos de muitos processadores separados, cada um com sua própria memória, ajuda a superar as limitações dos computadores com um único processador. Os **algoritmos paralelos**, que quebram um problema em diversos subproblemas que podem ser resolvidos ao mesmo tempo, podem ser projetados para resolver rapidamente problemas usando um computador com processadores múltiplos. Em um algoritmo paralelo, uma única linha de instrução controla a execução do algoritmo, enviando subproblemas para computadores diferentes, e dirige a entrada e a saída desses subproblemas para os processadores apropriados.

Quando o processamento paralelo é usado, um processador pode precisar da saída gerada por um outro processador. Conseqüentemente, estes processadores precisam estar interconectados. Precisamos usar um tipo apropriado de grafo para representar a rede de interconexão de processadores em um computador com processadores múltiplos. Na discussão a seguir, descreveremos os tipos mais comumente usados de redes de interconexão para processadores paralelos. O tipo de rede de interconexão usada para implementar um algoritmo paralelo particular depende das exigências de trocas de dados entre processadores, da velocidade desejada e, é claro, da disponibilidade de hardware.

A mais simples, mas mais cara, rede de interconexão de processadores inclui um link de mão dupla entre cada par de processadores. Essa rede pode ser representada por K_n, o grafo completo de n vértices, no qual existem n processadores. Entretanto, existem sérios problemas com este tipo de rede de interconexão, já que o número necessário de conexões é muito grande. Na realidade, o número de conexões diretas a um processador é limitado, de modo que quando existe um grande número de processadores, um processador não pode estar diretamente ligado a todos os outros. Por exemplo, quando existem 64 processadores, seriam necessárias $C(64,2) = 2016$ conexões e cada processador estaria diretamente conectado a 63 outros.

Por outro lado, talvez o modo mais simples de interconectar n processadores é usar um arranjo conhecido como **matriz linear** (ou **tabela linear**). Cada processador P_i, diferente de P_1 e P_n, está conectado a seus vizinhos P_{i-1} e P_{i+1} por um link de mão dupla. P_1 está conectado apenas a P_2, e P_n está conectado apenas a P_{n-1}. A matriz linear para seis processadores é mostrada na Figura 12. A vantagem de uma matriz linear é que cada processador tem, no máximo, duas conexões diretas a outros processadores. A desvantagem é que, às vezes, é necessário usar um grande número de links intermediários, chamados de **hops**, para os processadores compartilharem informação.

Uma **rede de malha** (ou uma **matriz bidimensional**) é uma rede de interconexão comumente usada. Em tal rede, o número de processadores é um quadrado perfeito, digamos $n = m^2$. Os n processadores são designados por $P(i, j)$, $0 \leq i \leq m-1$, $0 \leq j \leq m-1$. Links de mão dupla conectam o processador $P(i, j)$ com os seus quatro vizinhos, os processadores $P(i \pm 1, j)$ e $P(i, j \pm 1)$, contanto que estes sejam processadores em uma malha. (Observe que quatro processadores, nos cantos de uma malha, têm apenas dois processadores adjacentes, e outros processadores na fronteira têm apenas três vizinhos. Algumas vezes, é usada uma variante de uma rede de malha na qual todo processador tem exatamente quatro conexões; veja o Exercício 66 no final desta seção.) A rede de malha limita o número de links de cada processador. A comunicação entre

FIGURA 12 Uma Matriz Linear de Seis Processadores.

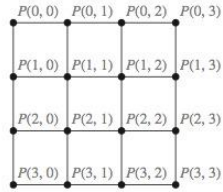


FIGURA 13 Uma Rede de Malha de 16 Processadores.

alguns pares de processadores exige $O(\sqrt{n}) = O(m)$ links intermediários. (Veja o Exercício 67 no final desta seção.) O grafo que representa uma rede de malha de 16 processadores é mostrado na Figura 13.

Um tipo importante de rede de interconexão é o hipercubo. Para tal rede, o número de processadores é uma potência de 2, $n = 2^m$. Os n processadores são designados por $P_0, P_1, \ldots, P_{n-1}$. Cada processador tem conexões de mão dupla com m outros processadores. O processador P_i está ligado aos processadores com índices cujas representações binárias diferem da representação binária de i em exatamente um bit. A rede hipercubo balanceia o número de conexões diretas para cada processador e o número de conexões intermediárias necessárias para que os processadores possam se comunicar. Muitos computadores foram construídos usando uma rede hipercubo, e muitos algoritmos paralelos foram projetados para serem usados em uma rede hipercubo. O grafo Q_m, o m-cubo, representa a rede hipercubo com $n = 2^m$ processadores. A Figura 14 mostra a rede hipercubo de oito processadores. (A Figura 14 mostra uma maneira diferente de desenhar Q_3 do que a que foi mostrada na Figura 6.) ◀

Novos Grafos a Partir de Outros

Algumas vezes precisamos apenas de parte de um grafo para resolver um problema. Por exemplo, podemos nos importar apenas com uma parte de uma grande rede de computadores que envolva os centros de computadores em Nova York, Denver, Detroit e Atlanta. Então, podemos ignorar os outros centros de computadores e todas as linhas telefônicas que não liguem dois destes quatro centros de computadores especificados. No modelo de grafo para a grande rede, podemos remover os vértices correspondentes aos outros centros de computadores, além dos quatro de interesse, e podemos remover todas as arestas incidentes com um vértice que tenha sido removido. Quando arestas e vértices são removidos de um grafo, sem remover extremidades de quaisquer arestas remanescentes, é obtido um grafo menor. Tal grafo é chamado de **subgrafo** do grafo original.

DEFINIÇÃO 6 Um *subgrafo de um grafo* $G = (V, E)$ é um grafo $H = (W, F)$, em que $W \subseteq V$ e $F \subseteq E$. Um subgrafo H de G é um *subgrafo próprio* de G se $H \neq G$.

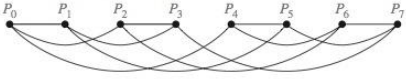


FIGURA 14 Uma Rede Hipercubo de Oito Processadores.

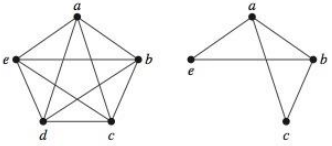


FIGURA 15 Um Subgrafo de K_5.

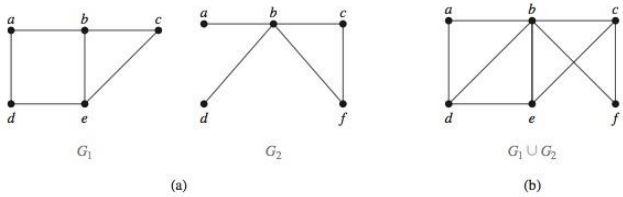


FIGURA 16 **(a) Os Grafos Simples G_1 e G_2; (b) Sua União $G_1 \cup G_2$.**

EXEMPLO 17 O grafo G mostrado na Figura 15 é um subgrafo de K_5. ◀

Dois ou mais grafos podem ser combinados de diversas maneiras. Um novo grafo que contém todos os vértices e arestas desses grafos é chamado de **união** dos grafos. Daremos uma definição mais formal da união de dois grafos simples.

DEFINIÇÃO 7 A *união* de dois grafos simples $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ é o grafo simples com conjunto de vértices $V_1 \cup V_2$ e conjunto de arestas $E_1 \cup E_2$. A união de G_1 e G_2 é indicada por $G_1 \cup G_2$.

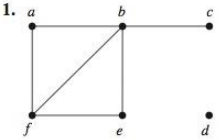
EXEMPLO 18 Encontre a união dos grafos G_1 e G_2 mostrados na Figura 16(a). ◀

Solução: O conjunto de vértices da união $G_1 \cup G_2$ é a união dos dois conjuntos de vértices, ou seja, $\{a, b, c, d, e, f\}$. O conjunto de arestas da união é a união dos dois conjuntos de arestas. A união está mostrada na Figura 16 (b).

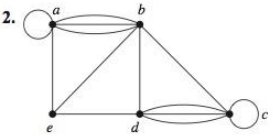
Exercícios

Nos exercícios 1 a 3, encontre o número de vértices, o número de arestas e o grau de cada vértice no grafo não orientado dado. Identifique todos os vértices isolados e pendentes.

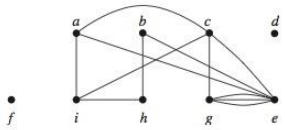
1.



2.



3.



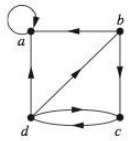
4. Encontre a soma dos graus dos vértices de cada grafo dos exercícios 1 a 3 e verifique que ela é igual a duas vezes o número de arestas do grafo.

5. É possível existir um grafo simples com 15 vértices, cada um de grau 5?

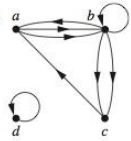
6. Mostre que a soma, sobre o conjunto de pessoas em uma festa, do número de pessoas a quem cada indivíduo deu a mão, é par. Suponha que ninguém dá a mão a si próprio.

Nos exercícios 7 a 9, determine o número de vértices e arestas e encontre o grau de entrada e o de saída de cada vértice para o multigrafo orientado dado.

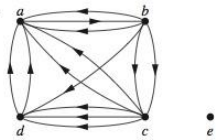
7.



8.



9.



10. Para cada um dos grafos nos exercícios 7 a 9, determine a soma dos graus de entrada dos vértices e a soma dos graus de saída dos vértices diretamente. Mostre que eles são ambos iguais ao número de arestas no grafo.

11. Construa o grafo não orientado subjacente ao grafo com arestas orientadas da Figura 2.

12. O que o grau de um vértice representa no grafo de relacionamento, no qual os vértices representam todas as pessoas do mundo? O que os vértices isolados e pendentes neste grafo representam? Em um estudo, foi estimado que o grau médio de um vértice neste grafo é 1 000. O que isso significa em termos do modelo?

13. O que o grau de um vértice representa em um grafo de colaboração? O que os vértices isolados e pendentes representam?

14. O que o grau de um vértice representa em um grafo de Hollywood? O que os vértices isolados e pendentes representam?

15. O que os graus de entrada e de saída de um vértice em um grafo de chamadas telefônicas, como descrito no Exemplo 7 da Seção 9.1, representam? O que o grau de um vértice na versão não orientada desse grafo representa?

16. O que os graus de entrada e de saída de um vértice em um grafo da Web, como descrito no Exemplo 8 da Seção 9.1, representam?

17. O que os graus de entrada e de saída de um vértice em um grafo orientado que modele um torneio *round-robin* representam?

18. Mostre que em um grafo simples com, pelo menos, dois vértices devem existir dois vértices que têm o mesmo grau.

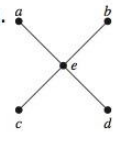
19. Use o Exercício 18 para mostrar que, em um grupo, devem existir duas pessoas que conhecem o mesmo número de outras pessoas do grupo.

20. Desenhe estes grafos.

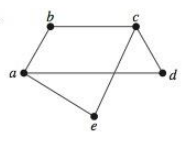
a) K_7 **b)** $K_{1,8}$ **c)** $K_{4,4}$
d) C_7 **e)** W_7 **f)** Q_4

Nos exercícios 21 a 25, determine se o grafo é bipartido. Você pode achar útil aplicar o Teorema 4 e responder à pergunta, determinando se é possível associar ou vermelho ou azul a cada vértice de modo que nenhum par de vértices adjacentes tenha a mesma cor designada.

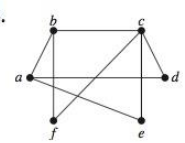
21.



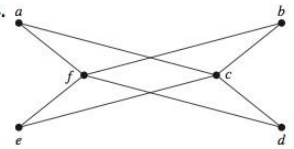
22.



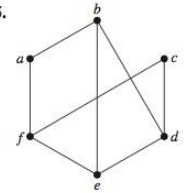
23.



24.



25.



26. Para qual valor de n estes grafos são bipartidos?

a) K_n **b)** C_n **c)** W_n **d)** Q_n

27. Suponha que uma nova companhia tenha cinco funcionários: Zamora, Agraharam, Smith, Chou e Macintyre. Cada funcionário assumirá uma de seis responsabilidades: planejamento, publicidade, vendas, propaganda, desenvolvimento e relações na indústria. Cada funcionário é capaz de fazer uma ou mais dessas tarefas: Zamora poderia fazer planejamento, vendas, propaganda ou relações na indústria; Agraharam poderia fazer planejamento ou desenvolvimento; Smith poderia fazer publicidade, vendas ou relações na indústria; Chou poderia fazer planejamento, vendas ou relações na indústria e Macintyre poderia fazer planejamento, publicidade, vendas ou relações na indústria.

a) Modele as habilidades desses funcionários usando um grafo bipartido.

b) Encontre uma atribuição de responsabilidades tal que a cada funcionário seja atribuída uma responsabilidade.

28. Suponha que existam cinco mulheres jovens e seis homens jovens em uma ilha. Cada mulher está disposta a casar com

alguns dos homens na ilha e cada homem está disposto a casar com qualquer mulher que esteja disposta a casar com ele. Suponha que Anna esteja disposta a casar com Jason, Larry e Matt; Barbara esteja disposta a casar com Kevin e Larry; Carol esteja disposta a casar com Jason, Nick e Oscar; Diane esteja disposta a casar com Jason, Larry, Nick e Oscar; e Elizabeth esteja disposta a casar com Jason e Matt.

a) Modele os casamentos possíveis na ilha usando um grafo bipartido.

b) Encontre uma correspondência entre as mulheres jovens e os homens jovens na ilha de modo que cada mulher jovem esteja associada a um homem jovem com o qual ela esteja disposta a casar.

29. Quantos vértices e quantas arestas estes grafos têm?

a) K_n **b)** C_n **c)** W_n

d) $K_{m,n}$ **e)** Q_n

A **seqüência de graus** de um grafo é a seqüência dos graus dos vértices do grafo em uma ordem não crescente. Por exemplo, a seqüência de graus do grafo G do Exemplo 1 desta seção é 4, 4, 4, 3, 2, 1, 0.

30. Encontre as seqüências de graus para cada um dos grafos nos exercícios 21 a 25.

31. Encontre a seqüência de graus para cada um dos seguintes grafos.

a) K_4 **b)** C_4 **c)** W_4

d) $K_{2,3}$ **e)** Q_3

32. Qual é a seqüência de graus do grafo bipartido $K_{m,n}$, em que m e n são inteiros positivos? Explique sua resposta.

33. Qual é a seqüência de graus de K_n, em que n é um inteiro positivo? Explique sua resposta.

34. Quantas arestas tem um grafo se sua seqüência de graus for 4, 3, 3, 2, 2? Desenhe tal grafo.

35. Quantas arestas tem um grafo se sua seqüência de graus for 5, 2, 2, 2, 2, 1? Desenhe tal grafo.

Uma seqüência $d_1, d_2, \ldots, d_n$ é chamada de **gráfica** se ela for a seqüência de graus de um grafo simples.

36. Determine se cada uma destas seqüências é gráfica. Para aquelas que forem, desenhe um grafo que tenha a seqüência de graus dada.

a) 5, 4, 3, 2, 1, 0 **b)** 6, 5, 4, 3, 2, 1

c) 2, 2, 2, 2, 2, 2 **d)** 3, 3, 3, 2, 2, 2

e) 3, 3, 2, 2, 2, 2 **f)** 1, 1, 1, 1, 1, 1

g) 5, 3, 3, 3, 3, 3 **h)** 5, 5, 4, 3, 2, 1

37. Determine se cada uma destas seqüências é gráfica. Para aquelas que forem, desenhe um grafo que tenha a seqüência de graus dada.

a) 3, 3, 3, 3, 2 **b)** 5, 4, 3, 2, 1

c) 4, 4, 3, 2, 1 **d)** 4, 4, 3, 3, 3

e) 3, 2, 2, 1, 0 **f)** 1, 1, 1, 1, 1

***38.** Suponha que $d_1, d_2, \ldots, d_n$ seja uma seqüência gráfica. Mostre que existe um grafo simples com vértices $v_1, v_2, \ldots, v_n$ tal que $\text{gr}(v_i) = d_i$ para $i = 1, 2, \ldots, n$, e v_1 é adjacente a $v_2, \ldots, v_{d_1+1}$.

***39.** Mostre que uma seqüência $d_1, d_2, \ldots, d_n$ de inteiros não negativos em uma ordem não crescente é uma seqüência gráfica se e somente se a seqüência obtida pela reordenação dos termos da seqüência $d_2 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n$ de modo que os termos estejam em uma ordem não crescente, for uma seqüência gráfica.

***40.** Use o Exercício 39 para construir um algoritmo recursivo para determinar se uma seqüência não crescente de inteiros positivos é gráfica.

41. Mostre que toda seqüência não crescente de inteiros não negativos com uma soma par de seus termos é a seqüência de graus de um pseudografo, isto é, um grafo não orientado no qual são permitidos laços. [*Dica*: Construa um gráfico primeiro adicionando tantos laços quanto possível em cada vértice. A seguir, inclua as arestas adicionais conectando vértices de grau ímpar. Explique por que esta construção funciona.]

42. Quantos subgrafos com, pelo menos, um vértice K_2 tem?

43. Quantos subgrafos com, pelo menos, um vértice K_3 tem?

44. Quantos subgrafos com, pelo menos, um vértice W_3 tem?

45. Desenhe todos os subgrafos deste grafo.

46. Seja G um grafo com v vértices e e arestas. Seja M o grau máximo dos vértices de G, e seja m o grau mínimo dos vértices de G. Mostre que

a) $2e/v \geq m$.

b) $2e/v \leq M$.

Um grafo simples é dito **regular** se todo vértice deste grafo tiver o mesmo grau. Um grafo regular é dito **n-regular** se cada vértice deste grafo tiver grau n.

47. Para quais valores de n estes grafos são regulares?

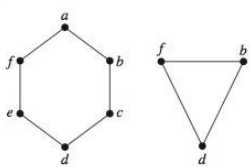
a) K_n **b)** C_n **c)** W_n **d)** Q_n

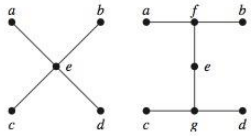
48. $K_{m,n}$ é regular para quais valores de m e n?

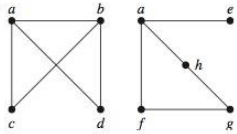
49. Quantos vértices tem um grafo regular de grau 4 com 10 arestas?

Nos exercícios 50 a 52, encontre a união do par de grafos simples dados. (Suponha que as arestas com as mesmas extremidades são as mesmas.)

50.



51. 

52. 

53. O **grafo complementar** $\overline{G}$ de um grafo simples G tem os mesmos vértices que G. Dois vértices são adjacentes em $\overline{G}$ se e somente se eles não forem adjacentes em G. Descreva cada um destes grafos.
a) $\overline{K_n}$ **b)** $\overline{K_{m,n}}$ **c)** $\overline{C_n}$ **d)** $\overline{Q_n}$

54. Se G for um grafo simples com 15 arestas e $\overline{G}$ tiver 13 arestas, quantos vértices G tem?

55. Se o grafo simples G tiver v vértices e e arestas, quantas arestas $\overline{G}$ tem?

56. Se a seqüência de graus de um grafo simples G for 4, 3, 3, 2, 2, qual é a seqüência de graus de $\overline{G}$?

57. Se a seqüência de graus de um grafo simples G for $d_1, d_2, \ldots, d_n$, qual é a seqüência de graus de $\overline{G}$?

***58.** Mostre que se G for um grafo simples bipartido com v vértices e e arestas, então $e \leq v^2/4$.

59. Mostre que se G for um grafo simples com n vértices, então a união de G e $\overline{G}$ é K_n.

***60.** Descreva um algoritmo para decidir se um grafo é bipartido com base no fato de que um grafo é bipartido se e somente se for possível colorir seus vértices com duas cores diferentes de modo que nenhum par de vértices que tenha a mesma cor seja adjacente.

O **reverso** de um grafo orientado $G = (V, E)$, indicado por G^{rev}, é o grafo orientado (V, F), no qual o conjunto F das arestas de G^{rev} é obtido invertendo o sentido de cada aresta em E.

61. Desenhe o reverso de cada um dos grafos dos exercícios 7 a 9 da Seção 9.1.

62. Mostre que $(G^{rev})^{rev} = G$ sempre que G for um grafo orientado.

63. Mostre que o grafo G é o seu próprio reverso se e somente se a relação associada a G (veja a Seção 8.3) for simétrica.

64. Mostre que se um grafo bipartido $G = (V, E)$ for n-regular para algum inteiro positivo n (veja o preâmbulo do Exercício 47) e (V_1, V_2) for uma bipartição de V, então $|V_1| = |V_2|$. Isto é, mostre que os dois conjuntos em uma bipartição do conjunto de vértices de um grafo n-regular devem conter o mesmo número de vértices.

65. Desenhe a rede de malha para 9 processadores paralelos interconectados.

66. Em uma variante de uma rede de malha para $n = m^2$ processadores interconectados, o processador $P(i, j)$ está conectado aos 4 processadores $P((i \pm 1) \bmod m, j)$ e $P(i, (j \pm 1) \bmod m)$, de modo que as conexões dão a volta nas arestas da malha. Desenhe esta variante da rede de malha para 16 processadores.

67. Mostre que todo par de processadores em uma rede de malha de $n = m^2$ processadores pode se comunicar usando $O(\sqrt{n}) = O(m)$ hops entre processadores conectados diretamente.

9.3 Representação de Grafos e Isomorfismo de Grafos

Introdução

Existem muitas maneiras úteis de representar grafos. Como veremos por todo este capítulo, quando trabalhamos com um grafo é útil ser capaz de escolher sua representação mais conveniente. Nesta seção, veremos como representar grafos de diversas maneiras diferentes.

Algumas vezes, dois grafos têm exatamente a mesma forma, no sentido que existe uma correspondência biunívoca entre seus conjuntos de vértices que preserva as arestas. Em tal caso, dizemos que os dois grafos são **isomorfos**. Determinar se dois grafos são isomorfos é um problema importante da teoria dos grafos que será estudado nesta seção.

Representação de Grafos

Uma forma de representar um grafo sem arestas múltiplas é listar todas as arestas desse grafo. Uma outra forma de representar um grafo sem arestas múltiplas é usar **listas de adjacência**, as quais especificam os vértices que são adjacentes a cada vértice do grafo.

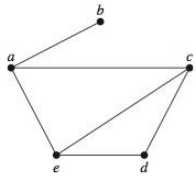


FIGURA 1 Um Grafo Simples.

TABELA 1 Uma Lista de Adjacência para um Grafo Simples.	
Vértice	*Vértices Adjacentes*
a	*b, c, e*
b	*a*
c	*a, d, e*
d	*c, e*
e	*a, c, d*

EXEMPLO 1 Use listas de adjacência para descrever o grafo simples dado na Figura 1.

Solução: A Tabela 1 lista aqueles vértices adjacentes a cada um dos vértices do grafo. ◀

EXEMPLO 2 Represente o grafo orientado mostrado na Figura 2 listando todos os vértices que são vértices finais de arestas que começam em cada vértice do grafo.

Solução: A Tabela 2 representa o grafo orientado mostrado na Figura 2. ◀

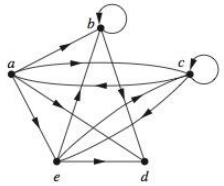


FIGURA 2 Um Grafo Orientado.

TABELA 2 Uma Lista de Adjacência para um Grafo Orientado.	
Vértice Inicial	*Vértice Final*
a	*b, c, d, e*
b	*b, d*
c	*a, c, e*
d	
e	*b, c, d*

FIGURA 3 Um Grafo Simples.

Matrizes de Adjacência

Executar algoritmos de grafos usando representação de grafos por meio de listas de arestas ou de listas de adjacências pode ser incômodo se existirem muitas arestas no grafo. Para simplificar os cálculos, os grafos podem ser representados usando matrizes. Dois tipos de matrizes comumente usados para representar grafos serão apresentados aqui. Um é baseado na adjacência dos vértices e o outro é baseado na incidência de vértices e arestas.

Links

Suponha que $G = (V, E)$ seja um grafo simples, em que $|V| = n$. Suponha que os vértices de G sejam listados arbitrariamente como $v_1, v_2, \ldots, v_n$. A **matriz de adjacência A** (ou $\mathbf{A}_G$) de G, com relação a esta listagem dos vértices, é a matriz zero-um n x n, com 1 como seu elemento (i, j) quando v_i e v_j forem adjacentes e 0 como seu elemento (i, j) quando eles não forem adjacentes. Em outras palavras, se sua matriz de adjacência é $\mathbf{A} = [a_{ij}]$, então

$$a_{ij} = \begin{cases} 1 & \text{se } \{v_i, v_j\} \text{ for uma aresta de } G, \\ 0 & \text{caso contrário.} \end{cases}$$

EXEMPLO 3 Use uma matriz de adjacência para representar o grafo mostrado na Figura 3.

Solução: Podemos ordenar os vértices como a, b, c, d. A matriz que representa este grafo é

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

◀

EXEMPLO 4 Desenhe um grafo com a matriz de adjacência

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

com relação à ordenação dos vértices a, b, c, d.

Solução: Um grafo com essa matriz de adjacência é mostrado na Figura 4. ◀

FIGURA 4
Um Grafo com a Matriz de Adjacência Dada.

Observe que uma matriz de adjacência de um grafo é baseada na ordem escolhida para os vértices. Por essa razão há tantas quanto $n!$ matrizes de adjacência diferentes para um grafo com n vértices, pois existem $n!$ ordenações diferentes de n vértices.

A matriz de adjacência de um grafo simples é simétrica, isto é, $a_{ij} = a_{ji}$, pois ambos os elementos são 1 quando v_i e v_j são adjacentes e ambos são 0 caso contrário. Além disso, como um grafo simples não tem nenhum laço, cada elemento a_{ii}, $i = 1, 2, 3, \ldots, n$, é 0.

As matrizes de adjacência também podem ser usadas para representar grafos não orientados com laços e com arestas múltiplas. Um laço em um vértice a_i é representado por um 1 na posição (i, i) da matriz de adjacência. Quando arestas múltiplas estiverem presentes, a matriz de adjacência não é mais uma matriz zero-um, pois o elemento (i, j) desta matriz é igual ao número de arestas que estão associadas a $\{a_i, a_j\}$. Todos os grafos não orientados, incluindo multigrafos e pseudografos, têm matrizes de adjacência simétricas.

EXEMPLO 5 Use uma matriz de adjacência para representar o pseudografo mostrado na Figura 5.

Solução: A matriz de adjacência que usa a ordenação dos vértices a, b, c, d é

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}.$$

◀

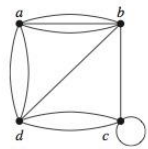


FIGURA 5
Um Pseudografo.

Usamos matrizes zero-um no Capítulo 8 para representar grafos orientados. A matriz para um grafo orientado $G = (V, E)$ tem um 1 em sua posição (i, j) se existir uma aresta de v_i para v_j, em que $v_1, v_2, \ldots, v_n$ é uma listagem arbitrária dos vértices do grafo orientado. Em outras palavras, se $\mathbf{A} = [a_{ij}]$ for a matriz de adjacência de um grafo orientado com relação a esta listagem de vértices, então

$$a_{ij} = \begin{cases} 1 & \text{se } (v_i, v_j) \text{ for uma aresta de } G, \\ 0 & \text{caso contrário.} \end{cases}$$

A matriz de adjacência de um grafo orientado não tem de ser simétrica, pois pode não haver uma aresta de a_j para a_i quando houver uma aresta de a_i para a_j.

As matrizes de adjacência também podem ser usadas para representar multigrafos orientados. Novamente, essas matrizes não são matrizes zero-um quando existirem arestas múltiplas no mesmo sentido conectando dois vértices. Na matriz de adjacência de um multigrafo orientado, a_{ij} é igual ao número de arestas que estão associados a (v_i, v_j).

PRÓS E CONTRAS ENTRE LISTAS DE ADJACÊNCIA E MATRIZES DE ADJACÊNCIA Quando um grafo simples contém relativamente poucas arestas, isto é, quando ele é **esparso**, normalmente é preferível usar listas de adjacência em vez de uma matriz de adjacência para representar o grafo. Por exemplo, se cada vértice tiver um grau que não exceda c, em que c é uma constante muito menor do que n, então cada lista de adjacência conterá c ou menos vértices. Portanto, não haverá mais do que cn itens em todas estas listas de adjacências. Por outro lado, a matriz de adjacência para o grafo tem n^2 elementos. Observe, entretanto, que a matriz de adjacência de um grafo esparso é uma **matriz esparsa**, isto é, uma matriz com poucos elementos não nulos, e existem técnicas especiais para representar e fazer cálculos com matrizes esparsas.

Supomos agora que um grafo simples seja **denso**, isto é, que ele contenha muitas arestas, tal como um grafo que contém mais da metade de todas as arestas possíveis. Neste caso, normalmente é preferível usar uma matriz de adjacência para representar o grafo em vez de usar listas de adjacência. Para perceber por que, comparamos a complexidade de determinar se uma aresta possível $\{v_i, v_j\}$ está presente. Usando uma matriz de adjacência, podemos determinar se essa aresta está presente examinando o elemento (i, j) da matriz. Este elemento é 1 se o grafo contém essa aresta e é 0 caso contrário. Conseqüentemente, precisamos fazer apenas uma comparação, ou seja, comparar este elemento com 0, para determinar se essa aresta está presente. Por outro lado, quando usamos listas de adjacências para representar o grafo, precisamos procurar a lista de vértices adjacentes tanto a v_i quanto a v_j para determinar se essa aresta está presente. Isto exige $\Theta(|V|)$ comparações quando muitas arestas estão presentes.

Matrizes de Incidência

Uma outra forma comum de representar grafos é usar **matrizes de incidência**. Seja $G = (V, E)$ um grafo não orientado. Suponha que $v_1, v_2, \ldots, v_n$ sejam os vértices e $e_1, e_2, \ldots, e_m$ sejam as arestas de G. Então a matriz de incidência com relação a esta ordem de V e E é a matriz n x m $\mathbf{M} = [m_{ij}]$, em que

$$m_{ij} = \begin{cases} 1 & \text{quando a aresta } e_j \text{ for incidente a } v_i, \\ 0 & \text{caso contrário.} \end{cases}$$

EXEMPLO 6 Represente o grafo mostrado na Figura 6 com uma matriz de incidência.

Solução: A matriz de incidência é

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}.$$

◀

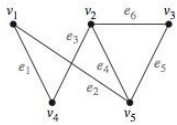


FIGURA 6 Um Grafo Não Orientado.

As matrizes de incidência também podem ser usadas para representar arestas múltiplas e laços. As arestas múltiplas são representadas na matriz de incidência usando colunas com elementos idênticos, pois estas arestas são incidentes ao mesmo par de vértices. Os laços são representados usando uma coluna com exatamente um elemento igual a 1, correspondendo ao vértice que é incidente deste laço.

EXEMPLO 7 Represente o pseudografo mostrado na Figura 7 usando uma matriz de incidência.

Solução: A matriz de incidência para este grafo é

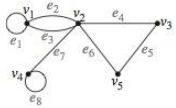


FIGURA 7
Um Pseudografo.

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}.$$

◀

Isomorfismo de Grafos

Freqüentemente precisamos saber se é possível desenhar dois grafos da mesma maneira. Por exemplo, em química, os grafos são usados para modelar compostos. Compostos diferentes podem ter a mesma fórmula molecular, mas diferirem na estrutura. Tais compostos serão representados por grafos que não podem ser desenhados da mesma maneira. Os grafos que representam compostos previamente conhecidos podem ser usados para determinar se um composto supostamente novo já foi estudado anteriormente.

Existe uma terminologia útil para grafos com a mesma estrutura.

DEFINIÇÃO 1 Os grafos simples $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ são *isomorfos* se existir uma função bijetora f de V_1 em V_2 com a propriedade que a e b são adjacentes em G_1 se e somente se $f(a)$ e $f(b)$ forem adjacentes em G_2, para todo a e b em V_1. Tal função f é chamada de *isomorfismo*.*

Em outras palavras, quando dois grafos simples são isomorfos, existe uma correspondência biunívoca entre os vértices de dois grafos que preserva a relação de adjacência. Isomorfismo de grafos simples é uma relação de equivalência. (Deixamos a verificação disso como Exercício 45 do final desta seção.)

EXEMPLO 8 Mostre que os grafos $G = (V, E)$ e $H = (W, F)$, apresentados na Figura 8, são isomorfos.

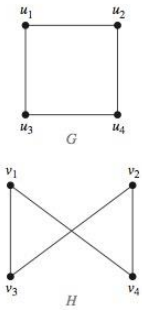


FIGURA 8 Os Grafos *G* e *H*.

Solução: A função f, com $f(u_1) = v_1$, $f(u_2) = v_4$, $f(u_3) = v_3$ e $f(u_4) = v_2$, é uma correspondência biunívoca entre V e W. Para ver que esta correspondência preserva a adjacência, observe que os vértices adjacentes em G são u_1 e u_2 , u_1 e u_3, u_2 e u_4, e u_3 e u_4, e cada um dos pares $f(u_1) = v_1$ e $f(u_2) = v_4$, $f(u_1) = v_1$ e $f(u_3) = v_3$, $f(u_2) = v_4$ e $f(u_4) = v_2$, e $f(u_3) = v_3$ e $f(u_4) = v_2$ são adjacentes em H. ◀

Freqüentemente é difícil determinar se dois grafos simples são isomorfos. Existem $n!$ correspondências biunívocas possíveis entre os conjuntos de vértices de dois grafos simples com n vértices. Testar cada uma dessas correspondências para ver se ela preserva a adjacência e a não-adjacência não é prático se n for um pouco grande.

Algumas vezes não é difícil mostrar que dois grafos não são isomorfos. Em particular, podemos mostrar que dois grafos não são isomorfos se pudermos encontrar uma propriedade que apenas um dos dois grafos tenha, mas que seja preservada por isomorfismo. Uma propriedade preservada por isomorfismo de grafos é chamada de **invariante do grafo**. Por exemplo, grafos simples isomorfos devem ter o mesmo número de vértices, pois existe uma correspondência biunívoca entre os conjuntos de vértices dos grafos.

* A palavra *isomorfismo* possui as raízes gregas *isos* para "igual" e *morphe* para "forma".

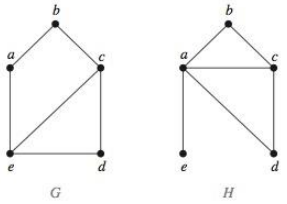


FIGURA 9 Os Grafos G e H.

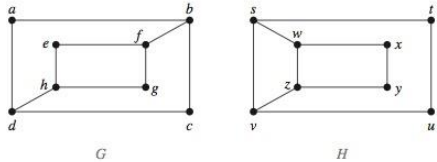


FIGURA 10 Os Grafos G e H.

Links

Grafos simples isomorfos também devem ter o mesmo número de arestas, pois a correspondência biunívoca entre os vértices estabelece uma correspondência biunívoca entre as arestas. Além disso, os graus dos vértices em grafos simples isomorfos devem ser os mesmos. Isto é, um vértice v de grau d em G deve corresponder a um vértice $f(v)$ de grau d em H, pois um vértice w em G é adjacente a v se e somente se $f(v)$ e $f(w)$ forem adjacentes em H.

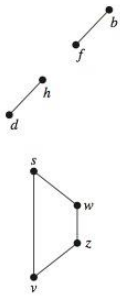
EXEMPLO 9 Mostre que os grafos na Figura 9 não são isomorfos.

Exemplos Extras

Solução: G e H têm, ambos, cinco vértices e seis arestas. Entretanto, H tem um vértice de grau 1, ou seja, e, enquanto que G não tem nenhum vértice de grau 1. Segue que G e H não são isomorfos. ◄

O número de vértices, o número de arestas e o número de vértices de cada grau são todos invariantes por isomorfismo. Se qualquer destas quantidades diferirem em dois grafos simples, estes grafos não podem ser isomorfos. Entretanto, quando estes invariantes forem os mesmos, isto não significa necessariamente que os dois grafos sejam isomorfos. Não existe nenhum conjunto útil de invariantes conhecidos atualmente que possam ser usados para determinar se grafos simples são isomorfos.

EXEMPLO 10 Determine se os grafos mostrados na Figura 10 são isomorfos.



Solução: Ambos os grafos G e H têm oito vértices e dez arestas. Ambos também têm quatro vértices de grau dois e quatro de grau três. Como todos estes invariantes coincidem, ainda é concebível que estes grafos sejam isomorfos.

Entretanto, G e H não são isomorfos. Para ver isto, observe que como $\text{gr}(a) = 2$ em G, a deve corresponder a t, u, x ou a y em H, pois estes são os vértices de grau dois em H. Entretanto, cada um destes quatro vértices em H é adjacente a um outro vértice de grau dois em H, o que não é verdade para a em G.

Uma outra maneira de ver que G e H não são isomorfos é observar que os subgrafos de G e H formados dos vértices de grau três e das arestas que os conectam devem ser isomorfos se estes dois grafos forem isomorfos (o leitor deve verificar isto). Entretanto, estes subgrafos, mostrados na Figura 11, não são isomorfos. ◄

FIGURA 11 Os Subgrafos de G e H Formados pelos Vértices de Grau Três e pelas Arestas que Os Conectam.

Para mostrar que uma função f do conjunto dos vértices do grafo G para o conjunto dos vértices do grafo H é um isomorfismo, precisamos mostrar que f preserva a presença e a ausência de arestas. Uma maneira útil de fazer isso é usar matrizes de adjacência. Em particular, para mostrar que f é um isomorfismo, podemos mostrar que a matriz de adjacência de G é a mesma que a matriz de adjacência de H, quando linhas e colunas forem escolhidas para corresponder às imagens indicadas por f dos vértices de G correspondentes a essas linhas e colunas na matriz de adjacência de G. Ilustramos como isto é feito no Exemplo 11.

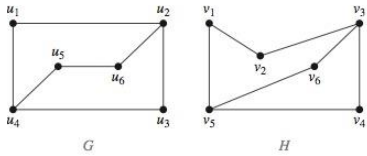


FIGURA 12 Grafos *G* e *H*.

EXEMPLO 11 Determine se os grafos G e H mostrados na Figura 12 são isomorfos.

Solução: G e H têm, ambos, seis vértices e sete arestas. Ambos têm quatro vértices de grau dois e dois vértices de grau três. Também é fácil ver que os subgrafos de G e H, que consistem em todos os vértices de grau dois e das arestas que os conectam, são isomorfos (como o leitor deve verificar). Como G e H coincidem em relação a todos estes invariantes, é razoável tentar encontrar um isomorfismo f.

Definiremos agora a função f e então determinaremos se ela é um isomorfismo. Como $\text{gr}(u_1) = 2$ e como u_1 não é adjacente a nenhum outro vértice de grau dois, a imagem de u_1 deve ser v_4 ou v_6, os únicos vértices de grau dois em H que não são adjacentes a nenhum vértice de grau dois. Escolhemos arbitrariamente $f(u_1) = v_6$. [Se descobrirmos que esta escolha não leva a um isomorfismo, tentaremos então $f(u_1) = v_4$.] Como u_2 é adjacente a u_1, as imagens possíveis de u_2 são v_3 e v_5. Tomamos arbitrariamente $f(u_2) = v_3$. Continuando desta forma, usando adjacência dos vértices e grau como guia, tomamos $f(u_3) = v_4$, $f(u_4) = v_5$, $f(u_5) = v_1$ e $f(u_6) = v_2$. Agora temos uma correspondência biunívoca entre o conjunto de vértices de G e o conjunto de vértices de H, ou seja, $f(u_1) = v_6$, $f(u_2) = v_3$, $f(u_3) = v_4$, $f(u_4) = v_5$, $f(u_5) = v_1$, $f(u_6) = v_2$. Para verificar se f preserva as arestas, examinamos a matriz de adjacência de G,

$$\mathbf{A}_G = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array} \begin{array}{c} \begin{array}{cccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array},$$

e a matriz de adjacência de H com as linhas e colunas escolhidas pelas imagens dos vértices correspondentes em G,

$$\mathbf{A}_H = \begin{array}{c} \\ v_6 \\ v_3 \\ v_4 \\ v_5 \\ v_1 \\ v_2 \end{array} \begin{array}{c} \begin{array}{cccccc} v_6 & v_3 & v_4 & v_5 & v_1 & v_2 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}.$$

Como $\mathbf{A}_G = \mathbf{A}_H$, segue que f preserva as arestas. Concluímos que f é um isomorfismo, de modo que G e H são isomorfos. Observe que se ocorresse de f não ser um isomorfismo, *não* teríamos estabelecido que G e H não são isomorfos, pois uma outra correspondência de vértices em G e H poderia ser um isomorfismo. ◀

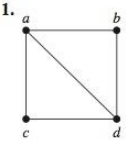
Links

O melhor algoritmo conhecido para determinar se dois grafos são isomorfos tem complexidade no tempo exponencial no pior caso (no número de vértices dos grafos). Entretanto, são conhecidos algoritmos com complexidade no tempo linear no caso médio que resolvem este problema, e há esperança que possa ser encontrado um algoritmo com complexidade no tempo polinomial no pior caso. O melhor algoritmo prático, chamado NAUTY, pode ser usado para determinar se dois grafos com até 100 vértices são isomorfos em menos de 1 segundo em um PC moderno. O software para o NAUTY pode ser baixado pela internet e experimentado.

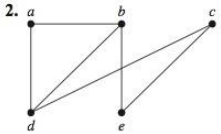
Exercícios

Nos exercícios 1 a 4, use uma lista de adjacência para representar o grafo dado.

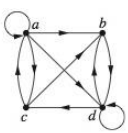
1.



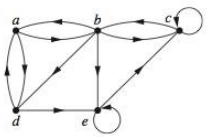
2.



3.



4.



5. Represente o grafo do Exercício 1 por uma matriz de adjacência.

6. Represente o grafo do Exercício 2 por uma matriz de adjacência.

7. Represente o grafo do Exercício 3 por uma matriz de adjacência.

8. Represente o grafo do Exercício 4 por uma matriz de adjacência.

9. Represente cada um destes grafos por uma matriz de adjacência.

a) K_4 **b)** $K_{1,4}$ **c)** $K_{2,3}$
d) C_4 **e)** W_4 **f)** Q_3

Nos exercícios 10 a 12, desenhe o grafo com a matriz de adjacência dada.

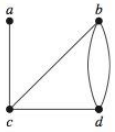
10. $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

11. $\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

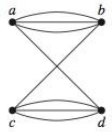
12. $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

Nos exercícios 13 a 15, represente o grafo dado usando uma matriz de adjacência.

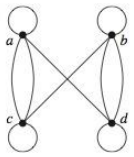
13.



14.



15.



Nos exercícios 16 a 18, desenhe um grafo não orientado representado pela matriz de adjacência dada.

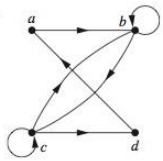
16. $\begin{bmatrix} 1 & 3 & 2 \\ 3 & 0 & 4 \\ 2 & 4 & 0 \end{bmatrix}$

17. $\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

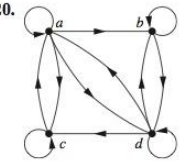
18. $\begin{bmatrix} 0 & 1 & 3 & 0 & 4 \\ 1 & 2 & 1 & 3 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$

Nos exercícios 19 a 21, encontre a matriz de adjacência do multigrafo orientado dado.

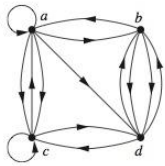
19.



20.



21.



Nos exercícios 22 a 24, desenhe o grafo representado pela matriz de adjacência dada.

22. $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ **23.** $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix}$ **24.** $\begin{bmatrix} 0 & 2 & 3 & 0 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 1 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}$

25. É verdade que toda matriz zero-um quadrada que é simétrica e tem zeros na diagonal é a matriz de adjacência de um grafo simples?

26. Use uma matriz de incidência para representar os grafos dos exercícios 1 e 2.

27. Use uma matriz de incidência para representar os grafos dos exercícios 13 a 15.

***28.** Qual é a soma dos elementos de uma linha de uma matriz de adjacência para um grafo não orientado? E para um grafo orientado?

***29.** Qual é a soma dos elementos de uma coluna de uma matriz de adjacência para um grafo não orientado? E para um grafo orientado?

30. Qual é a soma dos elementos de uma linha de uma matriz de incidência para um grafo não orientado?

31. Qual é a soma dos elementos de uma coluna de uma matriz de incidência para um grafo não orientado?

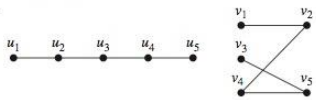
***32.** Encontre uma matriz de adjacência para cada um destes grafos.

a) K_n **b)** C_n **c)** W_n **d)** $K_{m,n}$ **e)** Q_n

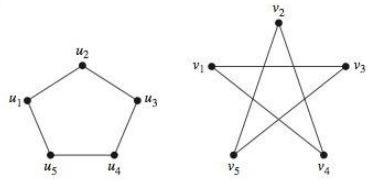
***33.** Encontre matrizes de incidência para os grafos nas partes (a) a (d) do Exercício 32.

Nos exercícios 34 a 44, determine se o par de grafos dado é isomorfo. Exiba um isomorfismo ou forneça um argumento rigoroso para que não exista nenhum.

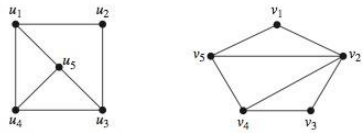
34.



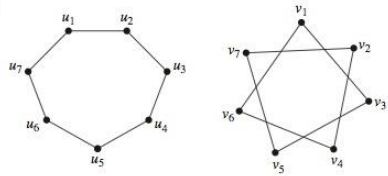
35.



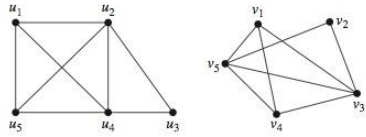
36.



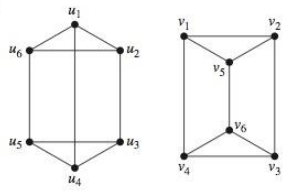
37.



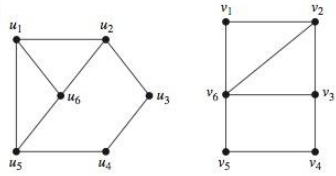
38.



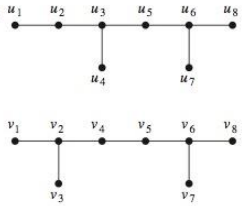
39.



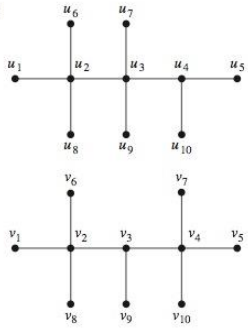
40.



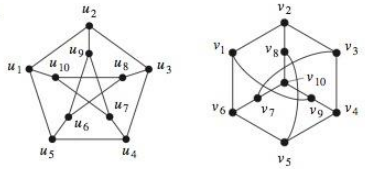
41.



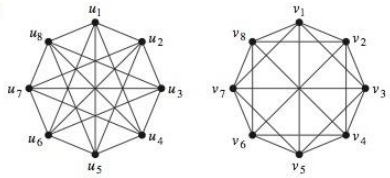
42.



43.



44.



45. Mostre que isomorfismo de grafos simples é uma relação de equivalência.

46. Suponha que G e H sejam grafos simples isomorfos. Mostre que seus grafos complementares $\overline{G}$ e $\overline{H}$ também são isomorfos.

47. Descreva a linha e a coluna de uma matriz de adjacência de um grafo que correspondem a um vértice isolado.

48. Descreva a linha de uma matriz de incidência de um grafo que corresponde a um vértice isolado.

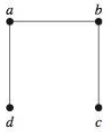
49. Mostre que os vértices de um grafo bipartido com dois ou mais vértices podem ser ordenados de modo que sua matriz de adjacência tenha a forma

$$\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{B} & \mathbf{0} \end{bmatrix},$$

em que os quatro elementos mostrados são blocos retangulares.

Um grafo simples G é dito **auto-complementar** se G e $\overline{G}$ forem isomorfos.

50. Mostre que este grafo é auto-complementar.



51. Encontre um grafo simples auto-complementar com cinco vértices.

***52.** Mostre que se G for um grafo simples auto-complementar com v vértices, então $v \equiv 0$ ou $1 \pmod 4$.

53. C_n é auto-complementar para quais inteiros n?

54. Quantos grafos simples não isomorfos existem com n vértices, em que n é

a) 2? **b)** 3? **c)** 4?

55. Quantos grafos simples não isomorfos existem com cinco vértice e três arestas?

56. Quantos grafos simples não isomorfos existem com seis vértices e quatro arestas?

57. Os grafos simples com as matrizes de adjacência a seguir são isomorfos?

a) $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

b) $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

58. Determine se os grafos sem laços com estas matrizes de incidência são isomorfos.

a) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

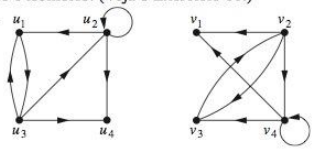
b) $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

59. Estenda a definição de grafos simples isomorfos a grafos não orientados com laços e arestas múltiplas.

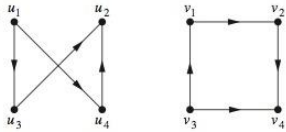
60. Defina isomorfismo de grafos orientados.

Nos exercícios 61 a 64, determine se o par de grafos orientados dado é isomorfo. (Veja o Exercício 60.)

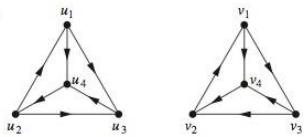
61.



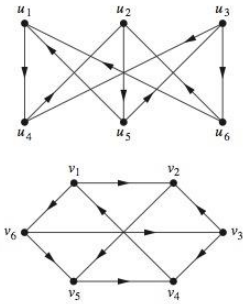
62.



63.



64.



65. Mostre que se G e H forem grafos orientados isomorfos, então os reversos de G e H (definido no preâmbulo do Exercício 61 da Seção 9.2) também são isomorfos.

66. Mostre que a propriedade de um grafo ser bipartido é um invariante por isomorfismo.

67. Encontre um par de grafos não isomorfos com a mesma seqüência de graus de modo que um grafo seja bipartido, mas o outro não.

***68.** Quantos grafos simples orientados não isomorfos existem com n vértices, em que n é

a) 2? **b)** 3? **c)** 4?

***69.** Qual é o produto da matriz de incidência por sua transposta para um grafo não orientado?

***70.** Quanto espaço de armazenamento é necessário para representar um grafo simples com v vértices e e arestas usando

a) listas de adjacência?

b) uma matriz de adjacência?

c) uma matriz de incidência?

Um **par do diabo** para um teste que se propõe a identificar o isomorfismo é um par de grafos não isomorfos para o qual o teste falha em mostrar que não são isomorfos.

71. Encontre um par do diabo para o teste que verifica a seqüência de graus (definida no preâmbulo do Exercício 30 da Seção 9.2) em dois grafos para ter certeza que são iguais.

9.4 Conectividade

Introdução

Muitos problemas podem ser modelados por caminhos formados percorrendo as arestas de um grafo. Por exemplo, o problema de determinar se uma mensagem pode ser enviada entre dois computadores usando links intermediários pode ser estudado com um modelo de grafo. Problemas de planejamento eficiente de rotas de entrega de correspondência, coleta de lixo, diagnóstico em redes de computador e assim por diante podem ser resolvidos usando modelos que envolvem caminhos em um grafo.

Caminhos

Informalmente, um **caminho** é uma seqüência de arestas que começa em um vértice de um grafo e continua de vértice em vértice ao longo de arestas do grafo.

Uma definição formal de caminhos e a terminologia relacionada estão dadas na Definição 1.

DEFINIÇÃO 1 Seja n um inteiro não negativo e G um grafo não orientado. Um *caminho de comprimento n* de u a v em G é uma seqüência de n arestas $e_1, \ldots, e_n$ de G, tal que e_1 está associada a $\{x_0, x_1\}$, e_2 está associada a $\{x_1, x_2\}$ e assim por diante, com e_n associada a $\{x_{n-1}, x_n\}$, em que $x_0 = u$ e $x_n = v$. Quando o grafo é simples, indicamos este caminho por sua seqüência de vértices x_0, $x_1, \ldots, x_n$ (pois listar estes vértices determina o caminho de forma única). O caminho é um *ciclo* se ele começa e termina no mesmo vértice, isto é, se $u = v$, e se tiver comprimento maior do que zero. Diz-se que o caminho ou ciclo *passa pelos* vértices $x_1, x_2, \ldots, x_{n-1}$ ou *percorre* as arestas $e_1, e_2, \ldots, e_n$. Um caminho ou ciclo é *simples* se ele não contém a mesma aresta mais de uma vez.

Quando não for necessário distinguir entre arestas múltiplas, indicaremos um caminho por $e_1, e_2, \ldots, e_n$, em que e_i está associada com $\{x_{i-1}, x_i\}$, para $i = 1, 2, \ldots, n$, por sua seqüência de vértices $x_0, x_1, \ldots, x_n$. Esta notação identifica um caminho apenas pelos vértices por onde ele passa. Pode existir mais de um caminho que passe por esta seqüência de vértices. Observe que um caminho de comprimento zero consiste em um único vértice.

Lembre-se: Existe uma variação considerável na terminologia a respeito dos conceitos expostos na Definição 1. Por exemplo, em alguns livros, o termo **passeio** é usado em vez de *caminho*, em que um passeio é definido como sendo uma seqüência alternada de vértices e arestas de um grafo, $v_0, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v_n$, em que v_{i-1} e v_i são as extremidades de e_i, para $i = 1, 2, \ldots, n$. Quando esta terminologia é usada, **passeio fechado** é usado em vez de *ciclo* para indicar um passeio que começa e termina no mesmo vértice, e **trilha** é usado para indicar um passeio que não tem arestas repetidas (substituindo o termo *caminho simples*). Quando esta terminologia é usada, a terminologia **caminho** é usada em geral para uma trilha sem vértices repetidos, conflitando com a terminologia da Definição 1. Por causa dessa variação na terminologia, você precisa ter certeza de qual conjunto de definições está sendo usado em um livro ou artigo particular quando ler sobre percorrer as arestas de um grafo. O texto [GrYe06] é uma boa referência para a terminologia alternativa descrita nesta observação.

EXEMPLO 1 No grafo simples mostrado na Figura 1, a, d, c, f, e é um caminho simples de comprimento 4, pois $\{a, d\}$, $\{d, c\}$, $\{c, f\}$ e $\{f, e\}$ são todas arestas. Entretanto, d, e, c e a não é um caminho, pois $\{e, c\}$ não é uma aresta. Observe que b, c, f, e, b é um ciclo de comprimento 4, pois $\{b, c\}$, $\{c, f\}$, $\{f, e\}$ e $\{e, b\}$ são arestas, e este caminho começa e termina em b. O caminho a, b, e, d, a, b, que é de comprimento 5, não é simples, pois contém a aresta $\{a, b\}$ duas vezes. ◀

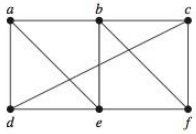


FIGURA 1 **Um Grafo Simples.**

Caminhos e ciclos em grafos orientados foram introduzidos no Capítulo 8. Fornecemos agora definições mais gerais.

DEFINIÇÃO 2 Seja n um inteiro não negativo e G um grafo orientado. Um *caminho* de comprimento n de u a v em G é uma seqüência de arestas $e_1, e_2, \ldots, e_n$ de G tal que e_1 está associada a (x_0, x_1), e_2 está associada a (x_1, x_2) e assim por diante, com e_n associada a (x_{n-1}, x_n), em que $x_0 = u$ e $x_n = v$. Quando não houver arestas múltiplas no grafo orientado, este caminho será indicado por sua seqüência de vértices $x_0, x_1, x_2, \ldots, x_n$. Um caminho de comprimento maior do que zero que comece e termine no mesmo vértice é chamado de *circuito* ou *ciclo*. Um caminho ou ciclo é dito *simples* se ele não contém a mesma aresta mais de uma vez.

Lembre-se: Terminologia diferente da dada na Definição 2 é usada com freqüência para os conceitos definidos lá. Em particular, a terminologia alternativa que usa *passeio*, *passeio fechado*, *trilha* e *caminho* (descrita na observação a seguir da Definição 1) pode ser usada para grafos orientados. Ver [GrYe06] para detalhes.

Observe que o vértice final de uma aresta em um caminho é o vértice inicial da próxima aresta no caminho. Quando não for necessário distinguir entre arestas múltiplas, indicaremos um caminho $e_1, e_2, \ldots, e_n$, em que e_i está associada a (x_{i-1}, x_i) para $i = 1, 2, \ldots, n$ por sua seqüência de vértices $x_0, x_1, \ldots, x_n$. A notação identifica um caminho apenas pela seqüência de vértices por onde ele passa. Pode existir mais de um caminho que passe por esta seqüência de vértices.

Caminhos representam informações úteis em muitos modelos de grafos, como ilustram os exemplos 2 a 4.

EXEMPLO 2

Caminhos em Grafos de Relacionamento Em um grafo de relacionamento, existe um caminho entre duas pessoas se existir uma cadeia de pessoas ligando estas pessoas, em que duas pessoas adjacentes nesta cadeia se conhecem. Por exemplo, na Figura 7 da Seção 9.1, existe uma cadeia de seis pessoas ligando Kamini e Ching. Muitos cientistas sociais têm conjeturado que quase todo par de pessoas no mundo está ligado por uma pequena cadeia de pessoas, contendo talvez cinco ou menos pessoas. Isto significaria que quase todo par de vértices no grafo de relacionamento que contém todas as pessoas do mundo está ligado por um caminho cujo comprimento não excede quatro. O filme *Seis Graus de Separação*, de John Guare, é baseado nesta noção. ◀

EXEMPLO 3

Caminhos em Grafos de Colaboração Em um grafo de colaboração, dois vértices a e b, que representam autores, estão ligados por um caminho sempre que existir uma seqüência de autores começando em a e terminando em b, tal que dois autores representados pelas extremidades de cada aresta escreveram um artigo em conjunto. No grafo de colaboração de todos os matemáticos, o **número de Erdös** de um matemático m (definido em termos das relações no Exercício Complementar 14 do Capítulo 8) é o comprimento do menor caminho entre m e o vértice que representa o matemático extremamente prolixo Paul Erdös (falecido em 1996). Isto é, o número de Erdös de um matemático é o comprimento do menor caminho que começa com Paul Erdös e termina com este matemático, em que cada par adjacente de matemáticos escreveu um artigo em conjunto. O número de matemáticos com cada número de Erdös no início de 2006, de acordo com o Projeto do Número de Erdös, está mostrado na Tabela 1. ◀

EXEMPLO 4

Caminhos em um Grafo de Hollywood Em um grafo de Hollywood (veja o Exemplo 4 da Seção 9.1), dois vértices a e b estão ligados quando existir uma cadeia de atores ligando a e b, em que dois atores adjacentes na cadeia atuaram em um mesmo filme. No grafo de Hollywood, o **número de Bacon** de um ator c é definido como o comprimento do menor caminho que liga c e

TABELA 1 O Número de Matemáticos com um Dado Número de Erdös (no início de 2006).

Número de Erdös	*Número de Pessoas*
0	1
1	504
2	6593
3	33605
4	83642
5	87760
6	40014
7	11591
8	3146
9	819
10	244
11	68
12	23
13	5

TABELA 2 O Número de Atores com um Dado Número de Bacon (no início de 2006).

Número de Bacon	*Número de Pessoas*
0	1
1	1902
2	160463
3	457231
4	111310
5	8168
6	810
7	81
8	14

o bem conhecido ator Kevin Bacon. À medida que novos filmes são feitos, inclusive os novos com Kevin Bacon, o número de Bacon de atores pode variar. Na Tabela 2, mostramos o número de atores com cada número de Bacon no início de 2006 usando dados do website Oracle of Bacon. ◀

Conexidade em Grafos Não Orientados

Quando uma rede de computadores tem a propriedade que cada par de computadores pode compartilhar informação, se as mensagens podem ser mandadas através de um ou mais computadores intermediários? Quando um grafo é usado para representar esta rede de computadores, no qual os vértices representam os computadores e as arestas representam os links de comunicação, esta questão se torna: quando sempre existe um caminho entre dois vértices de um grafo?

DEFINIÇÃO 3 Um grafo não orientado é dito *conexo* se existir um caminho entre qualquer par de vértices distintos do grafo.

Assim, quaisquer dois computadores de uma rede podem se comunicar se e somente se o grafo desta rede for conexo.

EXEMPLO 5 O grafo G_1 da Figura 2 é conexo, pois para cada par de vértices distintos existe um caminho entre eles (o leitor deve verificar isto). Entretanto, o grafo G_2 da Figura 2 não é conexo. Por exemplo, não existe caminho em G_2 entre os vértices a e d. ◀

Precisaremos do seguinte teorema no Capítulo 10.

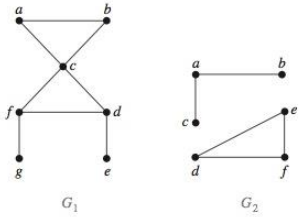


FIGURA 2 **Os Grafos G_1 e G_2.**

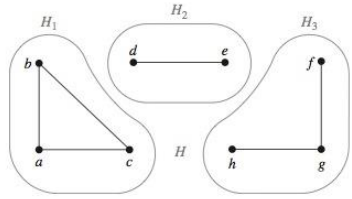


FIGURA 3 **O Grafo H e suas Componentes Conexas H_1, H_2 e H_3.**

TEOREMA 1 Existe um caminho simples entre todo par de vértices distintos de um grafo não orientado conexo.

Demonstração: Sejam u e v dois vértices distintos de um grafo não orientado conexo $G = (V, E)$. Como G é conexo, existe pelo menos um caminho entre u e v. Seja $x_0, x_1, \ldots, x_n$, em que $x_0 = u$ e $x_n = v$, a seqüência de vértices do caminho de menor comprimento. Este caminho de menor comprimento é simples. Para ver isso, suponha que ele não seja simples. Então $x_i = x_j$ para algum i e j com $0 \leq i < j$. Isto significa que existe um caminho de u a v de comprimento mais curto com a seqüência de vértices $x_0, x_1, \ldots, x_{i-1}, x_j, \ldots, x_n$ obtida retirando as arestas correspondentes à seqüência de vértices $x_i, \ldots, x_{j-1}$. ◁

Uma **componente conexa** de um grafo G é um subgrafo conexo de G que não é um subgrafo próprio de nenhum subgrafo conexo de G. Ou seja, uma componente conexa de um grafo G é um subgrafo conexo maximal de G. Um grafo G que não é conexo tem duas ou mais componentes conexas que são disjuntas e têm G como sua união.

Links

EXEMPLO 6 O que são as componentes conexas do grafo H mostrado na Figura 3?

Solução: O grafo H é a união de 3 subgrafos conexos disjuntos H_1, H_2 e H_3, mostrados na Figura 3. Esses três subgrafos são as componentes conexas de H. ◀

EXEMPLO 7 **Componentes Conexas de Grafos de Chamadas** Dois vértices x e y estão na mesma componente de um grafo de chamadas (veja o Exemplo 7 da Seção 9.1) quando existir uma seqüência de chamadas telefônicas começando em x e terminando em y. Quando um grafo de chamadas para chamadas telefônicas feitas durante um dia particular na rede AT&T foi analisado, descobriu-se que este grafo tinha 53 767 087 vértices, mais de 170 milhões de arestas e mais de 3,7 milhões de componentes conexas. A maioria destas componentes era pequena; aproximadamente três quartos consistiam em dois vértices representando pares de números telefônicos que ligaram apenas um para o outro. Este grafo tem uma enorme componente conexa com 44 989 297 vértices compreendendo mais de 80% do total. Além disso, todo vértice nesta componente pode ser ligado a qualquer outro vértice por uma cadeia de não mais de 20 chamadas. ◀

Links

Algumas vezes, a remoção de um vértice e de todas as arestas incidentes a ele produz um subgrafo com mais componentes conexas do que o grafo original. Tais vértices são chamados de **vértices de corte** (ou **pontos de articulação**). A remoção de um vértice de corte de um grafo conexo produz um subgrafo que não é conexo. Analogamente, uma aresta cuja remoção produz um grafo com mais componentes conexas do que o grafo original é chamada de **aresta de corte** ou **ponte**.

EXEMPLO 8 Encontre os vértices de corte e as arestas de corte no grafo G mostrado na Figura 4.

Solução: Os vértices de corte de G são b, c e e. A remoção de um desses vértices (e suas arestas adjacentes) desconecta o grafo. As arestas de corte são $\{a, b\}$ e $\{c, e\}$. A remoção de qualquer uma dessas arestas desconecta G. ◀

Conexidade em Grafos Orientados

Existem duas noções de conexidade em grafos orientados, dependendo de as orientações das arestas serem consideradas ou não.

DEFINIÇÃO 4 Um grafo orientado é *fortemente conexo* se existir um caminho de a a b e de b a a sempre que a e b forem vértices do grafo.

Para um grafo orientado ser fortemente conexo precisa existir uma seqüência de arestas orientadas de qualquer vértice no grafo para qualquer outro vértice. Um grafo orientado pode não ser fortemente conexo, mas ainda assim "estar inteiro". A Definição 5 torna esta noção precisa.

DEFINIÇÃO 5 Um grafo orientado é *fracamente conexo* se existir um caminho entre quaisquer dois vértices no grafo não orientado subjacente.

Ou seja, um grafo orientado é fracamente conexo se e somente se existir um caminho entre dois vértices quando as direções das arestas não forem consideradas. Claramente, qualquer grafo orientado fortemente conexo também é fracamente conexo.

EXEMPLO 9 Os grafos orientados G e H mostrados na Figura 5 são fortemente conexos? São fracamente conexos?

Solução: G é fortemente conexo, pois existe um caminho entre quaisquer dois vértices neste grafo orientado (o leitor deve verificar isto). Portanto, G também é fracamente conexo. O grafo H não é fortemente conexo. Não existe nenhum caminho orientado de a a b neste grafo. Entretanto, H é fracamente conexo, pois existe um caminho entre quaisquer dois vértices no grafo não orientado subjacente de H (o leitor deve verificar isto). ◀

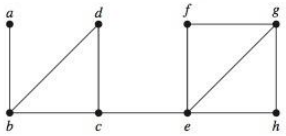


FIGURA 4 **O Grafo *G*.**

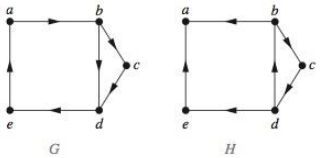


FIGURA 5 **Os Grafos Orientados *G* e *H*.**

Os subgrafos de um grafo orientado G que são fortemente conexos mas não estão contidos em subgrafos maiores fortemente conexos, isto é, os subgrafos fortemente conexos maximais, são chamados de **componentes fortemente conexas** ou **componentes fortes** de G.

EXEMPLO 10 O grafo H na Figura 5 tem três componentes fortemente conexas, que consiste no vértice a; no vértice e; e no grafo constituído dos vértices b, c e d e das arestas (b, c), (c, d) e (d, b). ◀

EXEMPLO 11 **As Componentes Fortemente Conexas de um Grafo da Web** O grafo da Web introduzido no Exemplo 8 da Seção 9.1 representa as páginas da Web como vértices e os links como arestas orientadas. Uma fotografia da Web em 1999 produziu um grafo da Web com mais de 200 milhões de vértices e 1,5 bilhão de arestas. (Veja [Br00] para detalhes.) O grafo não orientado subjacente desse grafo da Web não é conexo e tem uma componente conexa que inclui aproximadamente 90% dos vértices do grafo. O subgrafo do grafo orientado original correspondente a esta componente conexa do grafo não orientado subjacente (isto é, com os mesmos vértices e com todas as arestas orientadas ligando vértices neste grafo) tem uma componente fortemente conexa muito grande e muitas outras pequenas. A primeira é chamada de **componente fortemente conexa gigante (GSCC)** do grafo orientado. Uma página da Web desta componente pode ser atingida seguindo links que começam em qualquer outra página nesta componente. Descobriu-se que a GSCC no grafo da Web produzida por este estudo tinha mais de 53 milhões de vértices. Os vértices restantes da grande componente conexa do grafo não orientado representam três tipos diferentes de páginas da Web: páginas que podem ser acessadas a partir de uma página na GSCC, mas não têm uma conexão de volta para esta página seguindo uma série de links; páginas que estão ligadas de volta à página da GSCC seguindo uma série de links, mas não podem ser atingidas seguindo links nas páginas da GSCC; e páginas que não podem atingir páginas na GSCC e não podem ser atingidas a partir de páginas na GSCC seguindo uma série de links. Descobriu-se, neste estudo, que cada um destes três outros conjuntos tinha aproximadamente 44 milhões de vértices. (É bastante surpreendente que estes três conjuntos tenham aproximadamente o mesmo tamanho.) ◀

Links

Caminhos e Isomorfismo

Existem muitas maneiras de os caminhos e os ciclos ajudarem a determinar se dois grafos são isomorfos. Por exemplo, a existência de um ciclo simples de um comprimento particular é um invariante útil que pode ser usado para mostrar que dois grafos não são isomorfos. Além disso, os caminhos podem ser usados para construir aplicações que podem ser isomorfismos.

Como mencionamos, um invariante por isomorfismos útil para grafos simples é a existência de um ciclo simples de comprimento k, em que k é um inteiro positivo maior do que 2. (A demonstração de que isto é um invariante é deixada como Exercício 50 no final desta seção.) O Exemplo 12 ilustra como este invariante pode ser usado para mostrar que dois grafos não são isomorfos.

EXEMPLO 12 Determine se os grafos G e H mostrados na Figura 6 são isomorfos.

Solução: G e H têm, ambos, seis vértices e oito arestas. Cada um tem quatro vértices de grau três e dois vértices de grau dois. Assim, todos os três invariantes — o número de vértices, o número de arestas e os graus dos vértices — coincidem nos dois grafos. Entretanto, H tem um ciclo simples de comprimento três, a saber, v_1, v_2, v_6, v_1, enquanto G não tem nenhum ciclo simples de comprimento três, como pode ser determinado por inspeção (todos os ciclos simples em G têm comprimento pelo menos quatro). Como a existência de um ciclo simples de comprimento três é um invariante por isomorfismo, G e H não são isomorfos. ◀

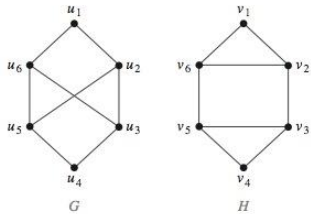


FIGURA 6 **Os Grafos *G* e *H*.**

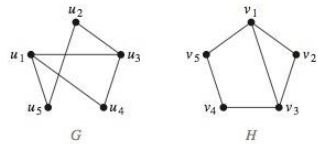


FIGURA 7 **Os Grafos *G* e *H*.**

Nós mostramos como a existência de um tipo de caminho, a saber, um ciclo simples de um comprimento particular, pode ser usado para mostrar que dois grafos não são isomorfos. Também podemos usar caminhos para encontrar aplicações que são isomorfismos em potencial.

EXEMPLO 13 Determine se os grafos G e H mostrados na Figura 7 são isomorfos.

Solução: G e H têm, ambos, cinco vértices e seis arestas, ambos têm dois vértices de grau três e três vértices de grau dois e ambos têm um ciclo simples de comprimento três, um ciclo simples de comprimento quatro e um ciclo simples de comprimento cinco. Como todos estes invariantes por isomorfismo coincidem, G e H podem ser isomorfos. Para encontrar um possível isomorfismo, podemos seguir caminhos que passem por todos os vértices de modo que os vértices correspondentes nos dois grafos tenham o mesmo grau. Por exemplo, ambos os caminhos u_1, u_4, u_3, u_2, u_5 em G e v_3, v_2, v_1, v_5, v_4 em H passam por todos os vértices do grafo; começam em um vértice de grau três; passam por vértices de grau dois, três e dois, respectivamente; e terminam em um vértice de grau dois. Seguindo esses caminhos através dos grafos, definimos a aplicação f por $f(u_1) = v_3, f(u_4) = v_2, f(u_3) = v_1, f(u_2) = v_5$ e $f(u_5) = v_4$. O leitor pode verificar que f é um isomorfismo, de modo que G e H são isomorfos, mostrando que f preserva as arestas ou mostrando que, com uma ordem apropriada dos vértices, as matrizes de adjacências de G e H são as mesmas. ◀

Contando Caminhos entre Vértices

O número de caminhos entre dois vértices em um grafo pode ser determinado usando sua matriz de adjacência.

TEOREMA 2 Seja G um grafo com matriz de adjacência $\mathbf{A}$ com relação à ordem $v_1, v_2, \ldots, v_n$ (com arestas orientadas ou não orientadas, e com arestas múltiplas e laços permitidos). O número de caminhos diferentes de comprimento r de v_i a v_j, em que r é um inteiro positivo, é igual ao elemento (i, j) de $\mathbf{A}^r$.

Demonstração: O teorema será demonstrado usando indução matemática. Seja G um grafo com matriz de adjacência $\mathbf{A}$ (considerando uma ordem $v_1, v_2, \ldots, v_n$ dos vértices de G). O número de caminhos de v_i a v_j de comprimento 1 é o elemento (i, j) de $\mathbf{A}$, pois este elemento é o número de arestas de v_i para v_j.

Suponha que o elemento (i, j) de $\mathbf{A}^r$ seja o número de caminhos diferentes de comprimento r de v_i a v_j. Esta é a hipótese de indução. Como $\mathbf{A}^{r+1} = \mathbf{A}^r \mathbf{A}$, o elemento (i, j) de $\mathbf{A}^{r+1}$ é igual a

$$b_{i1}\, a_{1j} + b_{i2} a_{2j} + \ldots + b_{in} a_{nj},$$

em que b_{ik} é o elemento (i, k) de $\mathbf{A}^r$. Pela hipótese de indução, b_{ik} é o número de caminhos de comprimento r de v_i a v_k.

Um caminho de comprimento $r + 1$ de v_i a v_j é formado de um caminho de comprimento r de v_i a algum vértice intermediário v_k e de uma aresta de v_k a v_j. Pela regra do produto para contagem, o número de tais caminhos é o produto do número de caminhos de comprimento r de v_i a v_k, ou seja, b_{ik}, e o número de arestas de v_k a v_j, ou seja, a_{kj}. Quando estes produtos são somados para todos os possíveis vértices intermediários v_k, o resultado desejado é obtido pela regra da soma para contagem. ◁

EXEMPLO 14 Quantos caminhos de comprimento quatro existem de a a d no grafo simples G da Figura 8?

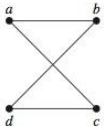


FIGURA 8 O Grafo *G*.

Solução: A matriz de adjacência de G (ordenando os vértices como a, b, c, d) é

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

Portanto, o número de caminhos de comprimento quatro de a a d é o elemento $(1, 4)$ de $\mathbf{A}^4$. Como

$$\mathbf{A}^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix},$$

Exemplos Extras existem exatamente oito caminhos de comprimento quatro de a a d. Por inspeção do grafo, vemos que a, b, a, b, d; a, b, a, c, d; a, b, d, b, d; a, b, d, c, d; a, c, a, b, d; a, c, a, c, d; a, c, d, b, d; e a, c, d, c, d são os oito caminhos de a a d. ◀

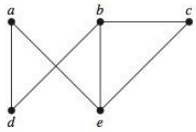
O Teorema 2 pode ser usado para encontrar o comprimento do menor caminho entre dois vértices de um grafo (veja o Exercício 46) e também pode ser usado para determinar se um grafo é conexo (veja os exercícios 51 e 52).

Exercícios

1. Cada uma das listas de vértices forma um caminho no grafo a seguir? Quais caminhos são simples? Quais são ciclos? Quais os comprimentos dos que são caminhos?

a) a, e, b, c, b **b)** a, e, a, d, b, c, a

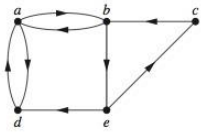
c) e, b, a, d, b, e **d)** c, b, d, a, e, c



2. Cada uma das listas de vértices forma um caminho no grafo a seguir? Quais caminhos são simples? Quais são ciclos? Quais os comprimentos dos que são caminhos?

a) a, b, e, c, b **b)** a, d, a, d, a

c) a, d, b, e, a **d)** a, b, e, c, b, d, a

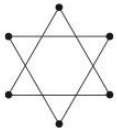


Nos exercícios 3 a 5, determine se o grafo dado é conexo.

3.

4.

5.



6. Quantas componentes conexas têm cada um dos grafos dos exercícios 3 a 5? Para cada grafo, encontre cada uma de suas componentes conexas.

7. O que representam as componentes conexas dos grafos de relacionamentos?

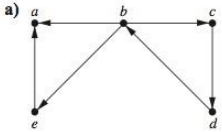
8. O que representam as componentes conexas de um grafo de colaboração?

9. Explique por que no grafo de colaboração dos matemáticos um vértice que representa um matemático está na mesma componente conexa do vértice que representa Paul Erdös se e somente se o matemático tem um número de Erdös finito.

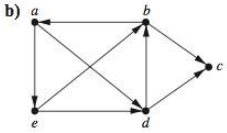
10. Em um grafo de Hollywood (veja o Exemplo 4 da Seção 9.1), quando o vértice que representa um ator está na mesma componente conexa que o vértice que representa Kevin Bacon?

11. Determine se cada um destes grafos é fortemente conexo e, se não for, se é fracamente conexo.

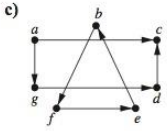
a)



b)

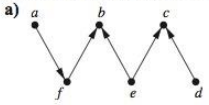


c)

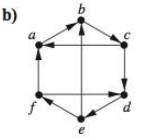


12. Determine se cada um destes grafos é fortemente conexo e, se não for, se é fracamente conexo.

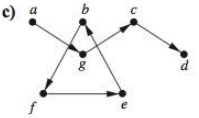
a)



b)



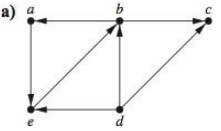
c)



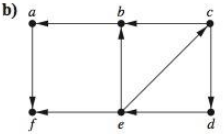
13. O que representam as componentes fortemente conexas de um grafo de chamadas telefônicas?

14. Encontre as componentes fortemente conexas de cada um destes grafos.

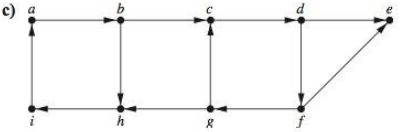
a)



b)

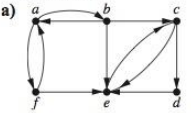


c)

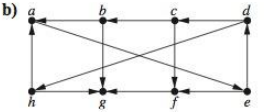


15. Encontre as componentes fortemente conexas de cada um destes grafos.

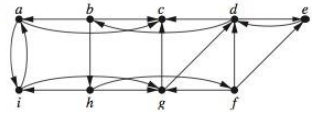
a)



b)



c)

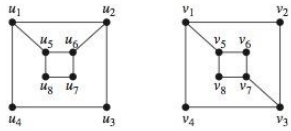


16. Mostre que todos os vértices visitados em um caminho orientado que conecta dois vértices na mesma componente fortemente conexa de um grafo orientado também estão nesta componente fortemente conexa.

17. Encontre o número de caminhos de comprimento n entre dois vértices diferentes de K_n se n for

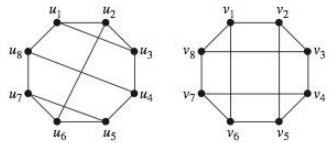
a) 2. **b)** 3. **c)** 4. **d)** 5.

18. Use caminhos ou para mostrar que esses grafos não são isomorfos ou para encontrar um isomorfismo entre esses grafos.



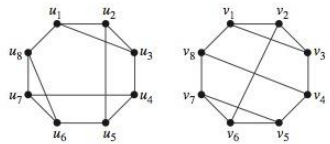
G *H*

19. Use caminhos ou para mostrar que estes grafos não são isomorfos ou para encontrar um isomorfismo entre eles.



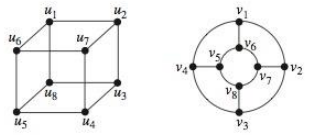
G *H*

20. Use caminhos ou para mostrar que estes grafos não são isomorfos ou para encontrar um isomorfismo entre eles.



G *H*

21. Use caminhos ou para mostrar que estes grafos não são isomorfos ou para encontrar um isomorfismo entre eles.



G *H*

22. Encontre o número de caminhos de comprimento n entre quaisquer dois vértices adjacentes de $K_{3,3}$ para os valores de n no Exercício 17.

23. Encontre o número de caminhos de comprimento n entre quaisquer dois vértices não adjacentes de $K_{3,3}$ para os valores de n no Exercício 17.

24. Encontre o número de caminhos entre c e d no grafo da Figura 1 de comprimento

a) 2. **b)** 3. **c)** 4. **d)** 5. **e)** 6. **f)** 7.

25. Encontre o número de caminhos de a a e no grafo orientado do Exercício 2 de comprimento

a) 2. **b)** 3. **c)** 4. **d)** 5. **e)** 6. **f)** 7.

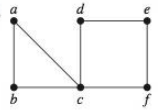
***26.** Mostre que todo grafo conexo com n vértices tem, no mínimo, $n - 1$ arestas.

27. Seja $G = (V, E)$ um grafo simples. Seja R uma relação em V que consiste em pares de vértices (u, v) tal que existe um caminho de u a v ou tal que $u = v$. Mostre que R é uma relação de equivalência.

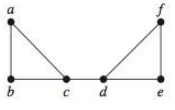
***28.** Mostre que em todo grafo simples existe um caminho de qualquer vértice de grau ímpar para algum outro vértice de grau ímpar.

Nos exercícios 29 a 31, encontre todos os vértices de corte do grafo dado.

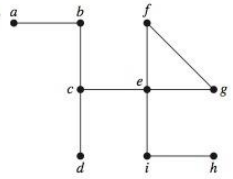
29.



30.



31.



32. Encontre todas as arestas de corte nos grafos dos exercícios 29 a 31.

***33.** Suponha que v seja uma extremidade de uma aresta de corte. Demonstre que v é um vértice de corte se e somente se este vértice não for pendente.

***34.** Mostre que um vértice c no grafo simples conexo G é um vértice de corte se e somente se existirem vértices u e v, ambos diferentes de c, tal que qualquer caminho entre u e v passe por c.

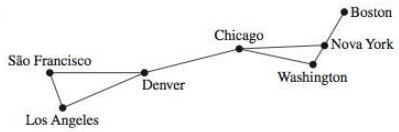
***35.** Mostre que um grafo simples com, pelo menos, dois vértices tem, pelo menos, dois vértices que não são vértices de corte.

***36.** Mostre que uma aresta em um grafo simples é uma aresta de corte se e somente se esta aresta não for parte de nenhum ciclo simples no grafo.

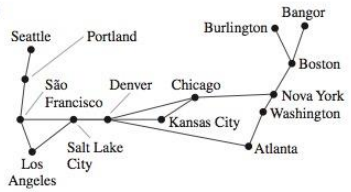
37. Um link de comunicação em uma rede deve ser munido de um link de segurança se a sua falha torna impossível que

alguma mensagem seja enviada. Para cada uma das redes de comunicação mostradas aqui em (a) e (b), determine aqueles links que deveriam ter links de segurança.

a)



b)



Uma **base de vértices** em um grafo orientado é um conjunto de vértices tal que existe um caminho para cada vértice no grafo orientado que não está no conjunto a partir de algum vértice neste conjunto e não existe nenhum caminho de qualquer vértice no conjunto para qualquer outro vértice no conjunto.

38. Encontre uma base de vértices para cada grafo orientado nos exercícios 7 a 9 da Seção 9.2.

39. Qual é o significado de uma base de vértices em um grafo de influência (descrito no Exemplo 3 da Seção 9.1)? Encontre uma base de vértices no grafo de influência daquele exemplo.

40. Mostre que se um grafo simples conexo G for a união dos grafos G_1 e G_2, então G_1 e G_2 têm, pelo menos, um vértice em comum.

***41.** Mostre que se um grafo simples G tiver k componentes conexas e estas componentes tiverem $n_1, n_2, ..., n_k$ vértices, respectivamente, então o número de arestas de G não excede

$$\sum_{i=1}^{k} C(n_i, 2).$$

***42.** Use o Exercício 41 para mostrar que um grafo simples com n vértices e k componentes conexas tem, no máximo, $(n-k)(n-k+1)/2$ arestas. [*Dica:* Primeiro mostre que

$$\sum_{i=1}^{k} n_i^2 \leq n^2 - (k-1)(2n-k),$$

em que n_i é o número de vértices na i-ésima componente conexa.]

***43.** Mostre que um grafo simples G com n vértices é conexo se ele tiver mais do que $(n-1)(n-2)/2$ arestas.

44. Descreva a matriz de adjacência de um grafo com n componentes conexas quando os vértices do grafo estiverem listados de modo que os vértices em cada componente conexa estejam listados sucessivamente.

45. Quantos grafos simples conexos não isomorfos existem com n vértices quando n é
a) 2? **b)** 3? **c)** 4? **d)** 5?

46. Explique como o Teorema 2 pode ser usado para determinar o comprimento do caminho mais curto de um vértice v a um vértice w em um grafo.

47. Use o Teorema 2 para encontrar o comprimento do caminho mais curto entre a e f no grafo da Figura 1.

48. Use o Teorema 2 para encontrar o comprimento do caminho mais curto de a a c no grafo orientado do Exercício 2.

☞ **49.** Sejam P_1 e P_2 dois caminhos simples entre os vértices u e v de um grafo simples G que não contém o mesmo conjunto de arestas. Mostre que existe um ciclo simples em G.

50. Mostre que a existência de um ciclo simples de comprimento k, em que k é um inteiro positivo maior do que 2, é um invariante por isomorfismo.

51. Explique como o Teorema 2 pode ser usado para determinar se um grafo é conexo.

52. Use o Exercício 51 para mostrar que o grafo G_1 na Figura 2 é conexo, enquanto que o grafo G_2 naquela figura não é conexo.

53. Mostre que um grafo simples G é bipartido se e somente se ele não tiver nenhum ciclo com um número ímpar de arestas.

54. Em um antigo desafio atribuído a Alcuin de York (735–804), um fazendeiro precisa atravessar um rio com um lobo, uma cabra e um repolho. O fazendeiro tem apenas um pequeno barco, que pode carregar somente ele e um único objeto (um animal ou um vegetal). Ele pode atravessar o rio várias vezes. Entretanto, se o fazendeiro estiver na outra margem, o lobo vai comer a cabra, e, do mesmo modo, a cabra vai comer o repolho. Podemos descrever cada estado listando o que está em cada margem. Por exemplo, podemos usar o par (FG, WC) para a situação na qual o fazendeiro e a cabra estão na primeira margem e o lobo e o repolho estão na outra margem. [O símbolo Ø é usado quando não houver nada em uma margem, de modo que ($FWGC$, Ø) é o estado inicial.]
a) Encontre todos os estados possíveis do desafio, nos quais nem o lobo e a cabra nem a cabra e o repolho são deixados na mesma margem sem o fazendeiro.
b) Construa um grafo tal que cada vértice desse grafo represente um estado permitido e os vértices que representam dois estados permitidos estejam ligados por uma aresta se for possível se mover de um estado para outro usando apenas uma viagem do barco.
c) Explique por que encontrar um caminho do vértice que representa ($FWGC$, Ø) para o vértice que representa (Ø, $FWGC$) resolve o desafio.
d) Encontre duas soluções diferentes do desafio, cada uma usando sete travessias.
e) Suponha que o fazendeiro precise pagar um pedágio de um dólar sempre que ele cruzar o rio com um animal. Qual solução do desafio o fazendeiro deveria usar para pagar um pedágio total mínimo?

***55.** Use um modelo de grafo e um caminho no seu grafo, como no Exercício 54, para resolver o **problema dos maridos ciumentos**. Dois casais casados, cada um com um marido e

uma esposa, querem cruzar um rio. Eles só podem usar um barco, que pode carregar uma ou duas pessoas de uma margem para outra. Cada marido é extremamente ciumento e não quer deixar sua esposa com o outro marido, nem no barco nem na margem. Como estas quatro pessoas podem atingir a margem oposta?

56. Suponha que você tenha uma jarra de três litros e uma jarra de cinco litros e que você pode encher qualquer uma das jarras em uma torneira de água, você pode esvaziar qualquer uma delas e você pode transferir água de uma para outra. Use um caminho em um modelo de grafo orientado para mostrar que você pode acabar com uma jarra que contém exatamente um litro. [*Dica:* use um par ordenado (a, b) para indicar quanta água tem em cada uma das jarras e represente os pares ordenados por vértices. Adicione arestas correspondentes às operações permitidas com as jarras.]

9.5 Caminhos Eulerianos e Hamiltonianos

Introdução

Podemos nos mover ao longo das arestas de um grafo começando em um vértice e retornando a ele percorrendo cada aresta do grafo exatamente uma vez? Do mesmo modo, podemos viajar ao longo das arestas de um grafo começando em um vértice e retornando a ele visitando cada vértice do grafo exatamente uma vez? Embora essas questões pareçam análogas, a primeira questão, que pergunta se um grafo tem um *ciclo euleriano*, pode ser facilmente respondida simplesmente examinando os graus dos vértices do grafo, enquanto a segunda questão, que pergunta se o grafo tem um *ciclo hamiltoniano*, é bastante difícil de resolver na maioria dos grafos. Nesta seção, estudaremos estas questões e discutiremos a dificuldade de resolvê-las. Embora ambas as questões tenham muitas aplicações em muitas áreas diferentes, ambas aparecem em desafios antigos. Aprenderemos sobre estes desafios antigos bem como as aplicações práticas modernas.

Caminhos e Ciclos Eulerianos

Links

A cidade de Königsberg, na Prússia (agora chamada Kaliningrad e parte da República Russa), era dividida em quatro sessões pelos braços do Rio Pregel. Estas quatro sessões incluíam as duas regiões das margens do Pregel, a ilha de Kneiphof e a região entre os dois braços. No século XVIII, sete pontes ligavam essas regiões. A Figura 1 descreve essas regiões e as pontes.

As pessoas da cidade davam longos passeios pela cidade aos domingos. Elas imaginavam se era possível começar em algum ponto da cidade, atravessar todas as pontes sem cruzar nenhuma ponte duas vezes e retornar ao ponto inicial.

O matemático suíço Leonhard Euler resolveu este problema. Sua solução, publicada em 1736, pode ser a primeira utilização da teoria dos grafos. Euler estudou este problema usando o multigrafo obtido quando as quatro regiões são representadas por vértices e as pontes por arestas. Este multigrafo está mostrado na Figura 2.

O problema de atravessar todas as pontes sem cruzar nenhuma ponte mais de uma vez pode ser reenunciado em termos deste modelo. A questão se torna: existe um ciclo simples neste multigrafo que contenha todas as arestas?

DEFINIÇÃO 1 Um *ciclo euleriano* em um grafo G é um ciclo simples que contém todas as arestas de G. Um *caminho euleriano* em G é um caminho simples que contém todas as arestas de G.

Os exemplos 1 e 2 ilustram o conceito de ciclos e caminhos eulerianos.

EXEMPLO 1 Quais dos grafos não orientados na Figura 3 têm um ciclo euleriano? Dos que não têm, quais têm um caminho euleriano?

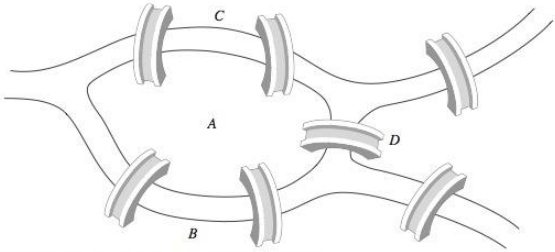


FIGURA 1 As Sete Pontes de Königsberg.

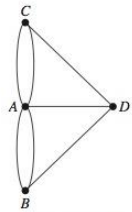


FIGURA 2 Modelo Multigrafo da Cidade de Königsberg.

Solução: O grafo G_1 tem um ciclo euleriano, por exemplo, a, e, c, d, e, b, a. Nenhum dos grafos G_2 ou G_3 tem um ciclo euleriano (o leitor deve verificar isto). Entretanto, G_3 tem um caminho euleriano, a saber, a, c, d, e, b, d, a, b. G_2 não tem um caminho euleriano (como o leitor deve verificar). ◀

EXEMPLO 2 Quais dos grafos orientados na Figura 4 têm um ciclo euleriano? Dos que não têm, quais têm um caminho euleriano?

Exemplos Extras

Solução: O grafo H_2 tem um ciclo euleriano, por exemplo, $a, g, c, b, g, e, d, f, a$. Nem H_1 nem H_3 tem um ciclo euleriano (como o leitor deve verificar). H_3 tem um caminho euleriano, a saber, c, a, b, c, d, b, mas H_1 não (como o leitor deve verificar). ◀

CONDIÇÕES NECESSÁRIAS E SUFICIENTES PARA CICLOS E CAMINHOS EULERIANOS Existem critérios simples para determinar se um multigrafo tem um ciclo euleriano ou um caminho euleriano. Euler descobriu-os quando resolveu o famoso problema das pontes de Königsberg. Vamos supor que todos os grafos discutidos nesta seção tenham um número finito de vértices e de arestas.

O que podemos dizer se um multigrafo conexo tiver um ciclo euleriano? O que podemos mostrar é que cada vértice deve ter grau par. Para fazer isto, observe primeiro que um ciclo euleriano começa em um vértice a e continua com uma aresta incidente em a, digamos $\{a, b\}$. A aresta $\{a, b\}$ contribui um para o gr(a). Cada vez que o ciclo passa por um vértice, ele contribui dois para o grau do vértice, pois o ciclo entra por uma aresta incidente ao vértice e sai por uma outra dessas arestas. Finalmente, o ciclo termina onde começou, contribuindo um para gr(a). Portanto, gr(a) deve ser par, pois o ciclo contribui um quando começa, um quando termina e dois cada vez que passa por a (se passar alguma vez). Um vértice diferente de a tem grau par, pois o ciclo contribui dois para seu grau cada vez que passa por esse vértice. Concluímos que se um grafo conexo tiver um ciclo euleriano, então todo vértice deve ter grau par.

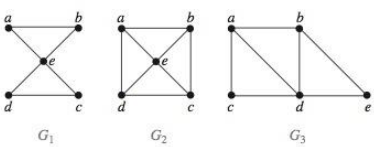


FIGURA 3 Os Grafos Não Orientados G_1, G_2 e G_3.

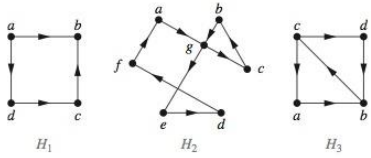


FIGURA 4 Os Grafos Orientados H_1, H_2 e H_3.

Esta condição necessária para a existência de um ciclo euleriano também é suficiente? Isto é, precisa existir um ciclo euleriano em um multigrafo conexo se todos os vértices tiverem grau par? Esta questão pode ser respondida afirmativamente com uma construção.

Suponha que G seja um multigrafo conexo com, pelo menos, dois vértices e que o grau de todos os vértices de G seja par. Formaremos um ciclo simples que começa em um vértice arbitrário a de G. Seja $x_0 = a$. Primeiro, escolhemos arbitrariamente uma aresta $\{x_0, x_1\}$ incidente em a, o que é possível já que G é conexo. Continuamos com a construção de um caminho simples $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$, adicionando arestas ao caminho até que não possamos mais adicionar outra aresta a ele. Isto ocorre quando atingimos um vértice no qual já tenhamos incluído todas as arestas incidentes nele no caminho. Por exemplo, no grafo G da Figura 5, começamos em a e escolhemos sucessivamente as arestas $\{a, f\}$, $\{f, c\}$, $\{c, b\}$ e $\{b, a\}$.

O caminho que construímos deve terminar, pois o grafo tem um número finito de arestas, de modo que temos certeza que eventualmente atingiremos um vértice para o qual não há nenhuma aresta disponível para adicionar ao caminho. O caminho começa em a com uma aresta da forma $\{a, x\}$ e agora mostraremos que ele deve terminar em a com uma aresta da forma $\{y, a\}$. Para ver que o caminho deve terminar em a, observe que cada vez que o caminho passa por um vértice de grau par, ele usa apenas uma aresta para chegar neste vértice, de modo que como o grau deve ser pelo menos dois, resta no mínimo uma aresta para o caminho deixar o vértice. Além disso, toda vez que chegamos em um vértice de grau par e o deixamos, existe um número par de arestas incidentes neste vértice que ainda não usamos em nossa trajetória. Conseqüentemente, à medida que formamos o caminho, toda vez que visitamos um vértice diferente de a, podemos deixá-lo. Isto significa que o caminho só pode terminar em a. A seguir, observe que o caminho que construímos pode usar todas as arestas do grafo ou pode não usar se tivermos retornado a a pela última vez antes de usar todas as arestas.

Um ciclo euleriano terá sido construído se todas as arestas tiverem sido usadas. Caso contrário, considere o subgrafo H obtido de G excluindo as arestas que já tenham sido usadas e os vértices que não são incidentes em nenhuma das arestas restantes. Quando excluímos o ciclo a, f, c, b, a do grafo da Figura 5, obtemos o subgrafo designado por H.

Como G é conexo, H tem, pelo menos, um vértice em comum com o ciclo que foi excluído. Seja w um desses vértices. (Em nosso exemplo c é o vértice.)

Todo vértice em H tem grau par (pois em G todos os vértices têm grau par e, para cada vértice, foram excluídos pares de arestas incidentes neste vértice para formar H). Observe que H pode não ser conexo. Começando em w, construa um caminho simples em H escolhendo arestas enquanto for possível, como foi feito em G. Este caminho deve terminar em w. Por exemplo, na Figura 5, c, d, e, c é um caminho em H. A seguir, forme um ciclo em G emendando o ciclo em H com o ciclo original em G (isto pode ser feito, pois w é um dos vértices deste ciclo). Quando isto é feito no grafo da Figura 5, obtemos o ciclo a, f, c, d, e, c, b, a.

Continue este processo até que todas as arestas tenham sido usadas. (Este processo deve acabar, já que existe apenas um número finito de arestas no grafo.) Isto produz um ciclo euleriano.

Links

LEONHARD EULER (1707–1783) Leonhard Euler era filho de um ministro calvinista dos arredores de Basel, na Suíça. Aos 13 anos, ele entrou na universidade de Basel, almejando uma carreira em teologia como seu pai queria. Na universidade, Euler foi orientado por Johann Bernoulli, da famosa família Bernoulli de matemáticos. Seu interesse e habilidades levaram-no a abandonar seus estudos teológicos e abraçar a matemática. Euler obteve seu título de mestre em filosofia com 16 anos de idade. Em 1727, Pedro, o Grande, convidou-o para se juntar à Academia em São Petersburgo. Em 1741, ele se mudou para a Academia de Berlim, onde permaneceu até 1766. Então ele voltou para São Petersburgo, onde permaneceu pelo resto de sua vida.

Euler foi incrivelmente prolífico, contribuindo em muitas áreas da matemática, incluindo a teoria dos números, combinatória e análise, bem como suas aplicações a áreas como música e arquitetura naval. Ele escreveu mais de 1,1 mil livros e artigos e deixou tanto trabalho não publicado que levou 47 anos após sua morte para todos serem publicados. Durante sua vida, seus artigos se acumulavam tão rapidamente que ele mantinha uma grande pilha de arquivos aguardando publicação. A Academia de Berlim publicava os artigos que estavam no topo da sua pilha, de modo que resultados posteriores eram, com freqüência, publicados antes dos resultados dos quais eles dependiam ou que os substituíam. Euler teve 13 filhos e era capaz de continuar seu trabalho enquanto uma ou duas crianças sentavam em seus joelhos. Ele ficou cego nos últimos 17 anos de sua vida, mas, por causa de sua memória fantástica, isto não diminuiu sua produção matemática. O projeto de publicar uma coletânea de seus trabalhos foi assumido pela Sociedade Suíça de Ciências Naturais, está em execução e exigirá mais de 75 volumes.

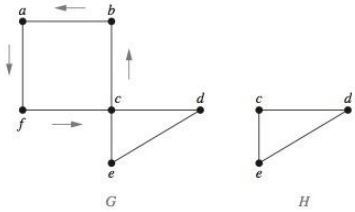


FIGURA 5 Construção de um Ciclo Euleriano em *G*.

A construção mostra que se todos os vértices de um multigrafo conexo tiverem grau par, então o grafo tem um ciclo euleriano.

Resumimos estes resultados no Teorema 1.

TEOREMA 1 Um multigrafo conexo com, pelo menos, dois vértices tem um ciclo euleriano se e somente se cada um de seus vértices tiver grau par.

Podemos agora resolver o problemas das pontes de Königsberg. Como o multigrafo que representa essas pontes, mostrado na Figura 2, tem quatro vértices de grau ímpar, ele não tem um ciclo euleriano. Não há nenhuma maneira de começar em um dado ponto, cruzar cada ponte exatamente uma vez e retornar ao ponto de partida.

O Algoritmo 1 apresenta o processo construtivo para encontrar ciclos eulerianos dado na discussão anterior ao Teorema 1. (Como os ciclos no processo são escolhidos arbitrariamente, há alguma ambigüidade. Não nos preocuparemos em remover essa ambigüidade especificando os passos do procedimento mais precisamente.)

ALGORITMO 1 Construção de Ciclos Eulerianos.

procedure *Euler*(*G*: multigrafo conexo com todos os vértices de grau par)
ciclo:= um ciclo em *G* começando em um vértice escolhido arbitrariamente com arestas adicionadas sucessivamente para formar um caminho que retorna a este vértice
H := *G* com as arestas deste ciclo removidas
while *H* tenha arestas
begin
  subciclo:= um ciclo em *H* que comece em um vértice de *H* que também seja uma extremidade de uma aresta do *ciclo*
  H:= *H* com as arestas do *subciclo* e todos os vértices isolados removidos
  ciclo:= *ciclo* com *subciclo* inserido no vértice apropriado
end {*ciclo* é um ciclo euleriano}

O Exemplo 3 mostra como caminhos e ciclos eulerianos podem ser usados para resolver um tipo de desafio.

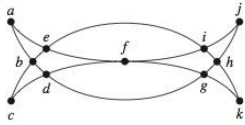


FIGURA 6 **As cimitarras de Mohammed.**

EXEMPLO 3 Muitos desafios lhe pedem para desenhar uma figura com um movimento contínuo sem levantar o lápis, de modo que nenhuma parte da figura seja retraçada. Podemos resolver tais desafios usando ciclos e caminhos eulerianos. Por exemplo, as *cimitarras de Mohammed*, mostradas na Figura 6, podem ser traçadas deste modo, no qual o desenho começa e termina no mesmo ponto?

Solução: Podemos resolver este problema, pois o grafo G mostrado na Figura 6 tem um ciclo euleriano. Ele tem tal ciclo porque todos os seus vértices têm grau par. Usaremos o Algoritmo 1 para construir um ciclo euleriano. Primeiro, formamos o ciclo $a, b, d, c, b, e, i, f, e, a$. Obtemos o subgrafo H excluindo as arestas deste ciclo e todos os vértices que se tornaram isolados quando estas arestas foram removidas. A seguir, formamos o ciclo $d, g, h, j, i, h, k, g, f, d$ em H. Depois de formar este ciclo, já teremos usado todas as arestas de G. A união deste novo ciclo ao primeiro ciclo no local apropriado produz o ciclo euleriano $a, b, d, g, h, j, i, h, k, g, f, d, c, b, e, i, f, e, a$. Este ciclo nos dá uma maneira de desenhar as cimitarras sem levantar o lápis nem retraçar parte da figura. ◀

Um outro algoritmo para construir ciclos eulerianos, chamado de algoritmo de Fleury, é descrito nos exercícios no final desta seção.

Mostraremos agora que um multigrafo conexo tem um caminho euleriano (e não um ciclo euleriano) se e somente se ele tiver exatamente dois vértices de grau ímpar. Primeiro, suponha que um multigrafo conexo de fato tenha um caminho euleriano de a para b, mas não um ciclo euleriano. A primeira aresta do caminho contribui um para o grau de a. É feita uma contribuição de dois para o grau de a toda vez que o caminho passa por a. A última aresta no caminho contribui um para o grau de b. Toda vez que o caminho passa por b, há uma contribuição de dois para o seu grau. Conseqüentemente, tanto a quanto b têm grau ímpar. Todos os outros vértices têm grau par, pois o caminho contribui com dois para o grau do vértice cada vez que passa por ele.

Considere agora a recíproca. Suponha que um grafo tenha exatamente dois vértices de grau ímpar, digamos a e b. Considere o grafo maior formado pelo grafo original com a adição de uma aresta $\{a, b\}$. Todos os vértices deste grafo maior têm grau par, de modo que existe um ciclo euleriano. A remoção desta nova aresta produz um caminho euleriano no grafo original. O Teorema 2 resume estes resultados.

TEOREMA 2 Um multigrafo conexo tem um caminho euleriano, mas não um ciclo euleriano se e somente se tiver exatamente dois vértices de grau ímpar.

EXEMPLO 4 Quais dos grafos mostrados na Figura 7 têm um caminho euleriano?

Solução: G_1 contém exatamente dois vértices de grau ímpar, a saber, b e d. Portanto, ele tem um caminho euleriano que deve ter b e d como suas extremidades. Um destes caminhos eulerianos é d, a, b, c, d, b. Analogamente, G_2 tem exatamente dois vértices de grau ímpar, a saber, b e d.

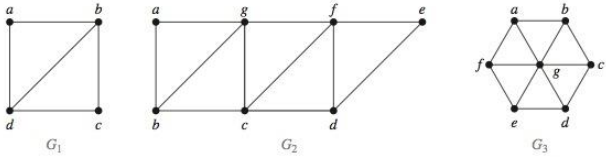


FIGURA 7 Três Grafos Não Orientados.

Assim, ele tem um caminho euleriano que deve ter b e d como extremidades. Um destes caminhos eulerianos é $b, a, g, f, e, d, c, g, b, c, f, d$. G_3 não tem nenhum caminho euleriano, pois tem seis vértices de grau ímpar. ◀

Voltando ao problema da Königsberg do século XVIII, é possível começar em algum ponto da cidade, percorrer todas as pontes e acabar em algum outro ponto da cidade? Esta questão pode ser respondida determinando-se se existe um caminho euleriano no multigrafo que representa as pontes de Königsberg. Como existem quatro vértices de grau ímpar neste multigrafo, não existe nenhum caminho euleriano, de modo que tal passeio é impossível.

Condições necessárias e suficientes para caminhos e ciclos eulerianos em grafos orientados são discutidas nos exercícios no final desta seção.

Links Caminhos e ciclos eulerianos podem ser usados para resolver muitos problemas práticos. Por exemplo, muitas aplicações pedem um caminho ou ciclo que percorra cada rua em uma vizinhança, cada estrada em uma rede de transporte, cada conexão em uma rede de serviços públicos ou cada link em uma rede de comunicações exatamente uma vez. Encontrar um caminho ou ciclo euleriano no modelo de grafo apropriado pode resolver tais problemas. Por exemplo, se um carteiro puder encontrar um caminho euleriano no grafo que representa as ruas que ele precisa cobrir, este caminho produz uma rota que percorre cada rua da rota exatamente uma vez. Se não existir um caminho euleriano, algumas ruas serão percorridas mais de uma vez. Este problema é conhecido como o *problema do carteiro chinês*, em homenagem a Guan Meigu, que o formulou em 1962. Veja [MiRo91] para mais informação sobre a solução do problema do carteiro chinês quando não existir nenhum caminho euleriano.

Entre outras áreas nas quais os ciclos e caminhos eulerianos são aplicados, estão o desenho de circuito, as redes de computadores de distribuição múltipla de dados e a biologia molecular, na qual os caminhos eulerianos são usados no seqüenciamento do DNA.

Caminhos e Ciclos Hamiltonianos

Links Desenvolvemos condições necessárias e suficientes para a existência de caminho e ciclos que contenham todas as arestas de um multigrafo exatamente uma vez. Podemos fazer o mesmo para caminhos e ciclos simples que contenham cada vértice do grafo exatamente uma vez?

DEFINIÇÃO 2 Um caminho simples em um grafo G que passe por todos os vértices exatamente uma vez é chamado de *caminho hamiltoniano* e um ciclo simples em um grafo G que passe pelos vértices exatamente uma vez é chamado de *ciclo hamiltoniano*. Isto é, o caminho simples $x_0, x_1, \ldots, x_{n-1}, x_n$ no grafo $G = (V, E)$ é um caminho hamiltoniano se $V = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ e $x_i \neq x_j$ para $0 \leq i < j \leq n$, e o ciclo simples $x_0, x_1, \ldots, x_{n-1}, x_n, x_0$ (com $n > 0$) é um ciclo hamiltoniano se $x_0, x_1, \ldots, x_{n-1}, x_n$ for um caminho hamiltoniano.

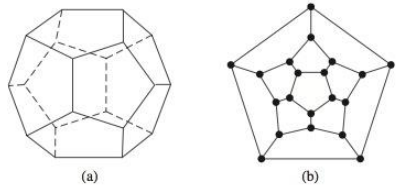


FIGURA 8 O Desafio de Hamilton "Uma Viagem ao Redor do Mundo".

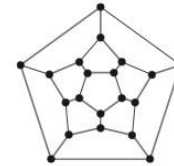


FIGURA 9 Uma Solução do desafio "Uma Viagem ao Redor do Mundo".

Esta terminologia vem de um jogo, chamado de *Icosian puzzle*, inventado em 1857 pelo matemático irlandês Sir William Rowan Hamilton. Ele consistia de um dodecaedro de madeira [um poliedro com 12 pentágonos regulares como faces, como mostrado na Figura 8(a)], com um pino em cada vértice do dodecaedro e barbante. Os 20 vértices do dodecaedro eram associados com diferentes cidades do mundo. O objetivo do jogo era começar em uma cidade e viajar ao longo das arestas do dodecaedro, visitando cada uma das outras 19 cidades exatamente uma vez, e terminar de volta na primeira cidade. O circuito percorrido era marcado usando os pinos e o barbante.

Como o autor não pode fornecer um sólido de madeira com pinos e barbante para cada leitor, consideraremos a questão equivalente: existe um ciclo no grafo mostrado na figura 8(b) que passe por todos os vértices exatamente uma vez? Isto resolve o desafio, pois este grafo é isomorfo ao grafo que consiste nos vértices e arestas do dodecaedro. Uma solução do desafio de Hamilton é mostrada na Figura 9.

EXEMPLO 5 Quais dos grafos simples na Figura 10 têm um ciclo hamiltoniano ou, caso não tenham, têm um caminho hamiltoniano?

Exemplos Extras

Solução: G_1 tem um ciclo hamiltoniano: a, b, c, d, e, a. Não existe nenhum ciclo hamiltoniano em G_2 (isto pode ser visto observando que qualquer ciclo que contenha todos os vértices deve conter a aresta $\{a, b\}$ duas vezes), mas G_2 tem um caminho hamiltoniano, a saber, a, b, c, d. G_3 não tem nem um ciclo nem um caminho hamiltoniano, pois qualquer caminho que contenha todos os vértices deve conter uma das arestas $\{a, b\}$, $\{e, f\}$ e $\{c, d\}$ mais de uma vez. ◀

Existe alguma maneira simples de determinar se um grafo tem um ciclo ou caminho hamiltoniano? Inicialmente, pode parecer que deva existir uma maneira fácil de determinar isso, pois existe uma maneira simples de responder a questão análoga de saber se um grafo tem um ciclo euleriano. Surpreendentemente, não existe nenhum critério necessário e suficientemente simples conhecido para a existência de ciclos hamiltonianos. Entretanto, são conhecidos muitos teoremas que dão condições suficientes para a existência de ciclos hamiltonianos.

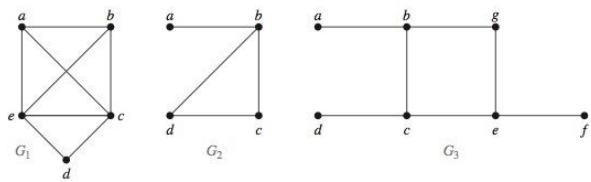


FIGURA 10 Três Grafos Simples.

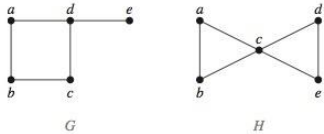


FIGURA 11 Dois Grafos Que Não Têm um Ciclo Hamiltoniano.

Além disso, certas propriedades podem ser usadas para mostrar que um grafo não tem nenhum ciclo hamiltoniano. Por exemplo, um grafo com um vértice de grau um não pode ter um ciclo hamiltoniano, pois em um ciclo hamiltoniano cada vértice é incidente a duas arestas. Além disso, se um vértice no grafo tiver grau dois, então ambas as arestas que são incidentes a este vértice devem ser parte de algum ciclo hamiltoniano. Observe também que quando um ciclo hamiltoniano estiver sendo construído e este ciclo tiver passado por um vértice, então todas as arestas remanescentes incidentes a este vértice, além das duas usadas no ciclo, podem ser desconsideradas. Ainda mais, um ciclo hamiltoniano não pode conter um ciclo menor dentro dele.

EXEMPLO 6 Mostre que nenhum dos grafos da Figura 11 tem um ciclo hamiltoniano.

Solução: Não existe nenhum ciclo hamiltoniano em G porque G tem um vértice de grau um, a saber, e. Considere H. Como os graus dos vértices a, b, d, e e são todos iguais a dois, cada aresta incidente a estes vértices devem ser parte de qualquer ciclo hamiltoniano. Agora é fácil ver que não existe nenhum ciclo hamiltoniano em H, pois qualquer ciclo hamiltoniano teria que conter quatro arestas incidentes a c, o que é impossível. ◀

Links

WILLIAM ROWAN HAMILTON (1805–1865) William Rowan Hamilton, o mais famoso cientista irlandês que já existiu, nasceu em 1805 em Dublin. Seu pai era um advogado de sucesso, sua mãe veio de uma família destacada por sua inteligência, e ele foi uma criança prodígio. Com a idade de três anos, ele já sabia ler muito bem e tinha dominado a aritmética avançada. Por causa de sua capacidade, ele foi mandado para viver com seu tio James, um lingüista destacado. Por volta dos oito anos, Hamilton tinha aprendido latim, grego e hebraico; aos dez, tinha aprendido também italiano e francês e começou seus estudos das línguas orientais, incluindo árabe, sânscrito e persa. Durante esse período, ele tinha orgulho de saber tantas línguas quanto sua idade. Aos dezessete, não mais interessado em aprender novas línguas, e tendo dominado o cálculo e muito da astronomia matemática, ele começou um trabalho original em ótica, e também descobriu um erro importante no trabalho de Laplace em mecânica celeste. Antes de ingressar no Trinity College, em Dublin, aos dezoito anos, Hamilton não tinha freqüentado a escola; em vez disso, ele recebeu aulas particulares. Em Trinity, ele era um estudante destacado tanto em ciências quanto nos clássicos. Antes de receber seu diploma, por causa de seu talento, foi nomeado o Astrônomo Real da Irlanda, vencendo diversos astrônomos famosos para o posto. Ele manteve esta posição até sua morte, vivendo e trabalhando no observatório de Dunsink, nos arredores de Dublin. Hamilton fez contribuições importantes à ótica, álgebra abstrata e dinâmica. Inventou objetos algébricos chamados de quatérnios como exemplo de um sistema não comutativo. Ele descobriu uma maneira apropriada de multiplicar os quatérnios enquanto caminhava ao longo de um canal de Dublin. Em seu excitamento, ele gravou a fórmula na pedra de uma ponte que cruzava o canal, um ponto marcado atualmente por uma placa. Mais tarde, Hamilton permaneceu obcecado pelos quatérnios, trabalhando para aplicá-los a outras áreas da matemática, em vez de se mover para outras áreas de pesquisa.

Em 1857, Hamilton inventou o "Icosian Game", baseado em seu trabalho em álgebra não comutativa. Ele vendeu a idéia por 25 libras para um comerciante de jogos e quebra-cabeças. (Como o jogo nunca vendeu bem, este acabou sendo um mau investimento para o comerciante.) O "Dodecaedro do Viajante", também chamado de "Uma Viagem ao Redor do Mundo", o jogo que descrevemos nesta seção, é uma variante daquele jogo.

Hamilton casou com seu terceiro amor em 1833, mas seu casamento não funcionou bem, pois sua esposa, uma semi-inválida, não foi capaz de lidar com seus afazeres domésticos. Ele sofreu de alcoolismo e viveu recluso durante as duas últimas décadas de sua vida. Morreu de gota em 1865, deixando uma grande quantidade de artigos com pesquisas não publicadas. Misturado com esses artigos, havia um grande número de pratos de jantar, muitos com restos de costeletas dissecadas, não comidas.

EXEMPLO 7 Mostre que K_n tem um ciclo hamiltoniano sempre que $n \geq 3$.

Solução: Podemos formar um ciclo hamiltoniano em K_n começando em qualquer vértice. Tal circuito pode ser construído visitando os vértices em qualquer ordem que escolhamos, contanto que o caminho comece e termine no mesmo vértice e visite cada um dos outros vértices exatamente uma vez. Isto é possível porque existem arestas em K_n entre quaisquer dois vértices. ◀

Embora não seja conhecida nenhuma condição necessária e suficiente para a existência de ciclos hamiltonianos, várias condições suficientes foram encontradas. Observe que quanto mais arestas um grafo tiver, mais provável a existência de um ciclo hamiltoniano. Além disso, adicionar arestas (mas não vértices) a um grafo com um ciclo hamiltoniano produz um grafo com o mesmo ciclo hamiltoniano. Assim, conforme adicionarmos arestas a um grafo, especialmente se prestarmos atenção para adicionar arestas a cada vértice, tornaremos cada vez mais provável que exista um ciclo hamiltoniano neste grafo. Conseqüentemente, esperaríamos que existissem condições suficientes para a existência de ciclos hamiltonianos que dependam de os graus dos vértices serem suficientemente grandes. Enunciamos aqui duas das mais importantes condições suficientes. Estas condições foram descobertas por Gabriel A. Dirac em 1952, e por Oystein Ore em 1960.

TEOREMA 3 TEOREMA DE DIRAC Se G for um grafo simples com n vértices com $n \geq 3$ tal que o grau de todo vértice em G seja, pelo menos, $n/2$, então G tem um ciclo hamiltoniano.

TEOREMA 4 TEOREMA DE ORE Se G for um grafo simples com n vértices com $n \geq 3$ tal que $\text{gr}(u) + \text{gr}(v) \geq n$ para cada par de vértices não adjacentes u e v em G, então G tem um ciclo hamiltoniano.

A demonstração do Teorema de Ore é esboçada no Exercício 65 no final desta seção. O Teorema de Dirac pode ser deduzido do Teorema de Ore, pois as condições do Teorema de Dirac implicam aquelas do Teorema de Ore.

Ambos os Teoremas de Ore e de Dirac fornecem condições suficientes para que um grafo conexo simples tenha um ciclo hamiltoniano. Entretanto, esses teoremas não fornecem condições necessárias para a existência de um ciclo hamiltoniano. Por exemplo, o grafo C_5 tem um ciclo hamiltoniano, mas não satisfaz as hipóteses nem do Teorema de Ore nem do Teorema de Dirac, como o leitor pode verificar.

O melhor algoritmo conhecido para encontrar um ciclo hamiltoniano em um grafo, ou determinar que tais ciclos não existam, tem complexidade no tempo exponencial no pior caso (no número de vértices do grafo). Encontrar um algoritmo que resolva este problema com complexidade no tempo polinomial no pior caso seria um feito notável, pois foi mostrado que este problema é NP-completo (veja a Seção 3.3). Conseqüentemente, a existência de tal algoritmo implicaria que muitos outros problemas aparentemente intratáveis poderiam ser resolvidos usando algoritmos com complexidade no tempo polinomial no pior caso.

Os caminhos e ciclos hamiltonianos podem ser usados para resolver problemas práticos. Por exemplo, muitas aplicações pedem um caminho ou ciclo que visite cada intersecção de estradas em uma cidade, cada lugar onde tubulações se interceptam em uma rede de serviços públicos, ou cada nó em uma rede de comunicação, exatamente uma vez. Encontrar um caminho ou ciclo hamiltoniano no modelo de grafo apropriado pode resolver tais problemas. O famoso **problema do caixeiro-viajante** pede a menor rota que um caixeiro-viajante deveria tomar para visitar um conjunto de cidades. Este problema se reduz a encontrar um ciclo hamiltoniano em um grafo completo tal que o peso total de suas arestas seja tão pequeno quanto possível. Retornaremos a esta questão na Seção 9.6.

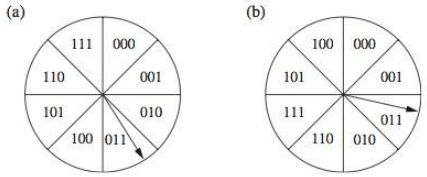


FIGURA 12 Conversão da Posição de um Ponteiro para a Forma Digital.

Descreveremos agora uma aplicação menos óbvia dos ciclos hamiltonianos em codificação.

EXEMPLO 8 Códigos de Gray A posição de um ponteiro em rotação pode ser representada na forma digital. Uma maneira de fazer isso é dividir o círculo em 2^n arcos de comprimentos iguais e associar uma seqüência de bits de comprimento n a cada arco. Duas maneiras de fazer isso usando seqüências de bits de comprimento 3 são mostradas na Figura 12.

A representação digital da posição do ponteiro pode ser determinada usando um conjunto de n contatos. Cada contato é usado para ler um bit na representação digital da posição. Isso é ilustrado na Figura 13 para as duas associações da Figura 12.

Quando o ponteiro está próximo da fronteira de dois arcos, pode ocorrer um erro na leitura de sua posição. Isto pode resultar em um grande erro na seqüência de bits lida. Por exemplo, no esquema de código da Figura 12 (a), se um pequeno erro for feito na determinação da posição do

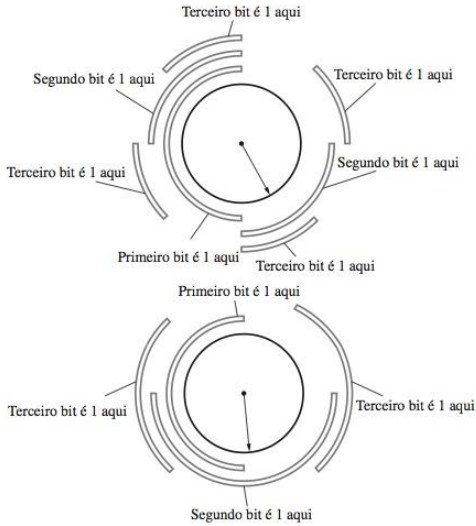


FIGURA 13 A Representação Digital da Posição do Ponteiro.

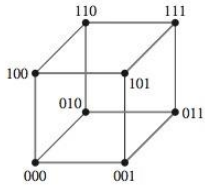


FIGURA 14 Um Ciclo Hamiltoniano para Q_3.

ponteiro, a cadeia de bits 100 será lida em vez da 011. Todos os três bits estarão incorretos! Para minimizar o efeito de um erro na determinação da posição do ponteiro, a associação das seqüências de bits aos 2^n arcos deveria ser feita de modo que apenas um bit fosse diferente na seqüência de bits representada por arcos adjacentes. Esta é exatamente a situação no esquema de código da Figura 12 (b). Um erro na determinação da posição do ponteiro fornece a cadeia de bits 010 em vez da 011. Apenas um bit está errado.

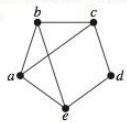
Links Um **código de Gray** é uma designação dos arcos do círculo tal que arcos adjacentes são designados por cadeias de bits que diferem exatamente em um bit. A associação na Figura 12 (b) é um código de Gray. Podemos encontrar um código de Gray listando todas as seqüências de bits de comprimento n de tal forma que cada seqüência difira em exatamente uma posição da seqüência de bits precedente, e a última seqüência difira da primeira seqüência em exatamente uma posição. Podemos modelar este problema usando o n-cubo Q_n. O que é necessário para resolver este problema é um ciclo hamiltoniano em Q_n. Tais ciclos hamiltonianos são encontrados facilmente. Por exemplo, um ciclo hamiltoniano para Q_3 é mostrado na Figura 14. A seqüência de cadeias de bits que diferem exatamente em um bit produzida por este ciclo hamiltoniano é 000, 001, 011, 010, 110, 111, 101, 100.

Os códigos de Gray têm este nome em homenagem a Frank Gray, que os inventou na década de 1940 no AT&T Bell Laboratories para minimizar o efeito de erros na transmissão de sinais digitais. ◀

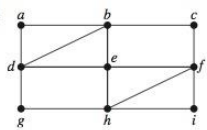
Exercícios

Nos exercícios 1 a 8, determine se o grafo dado tem um ciclo euleriano. Construa um ciclo desses quando existir. Se não existir nenhum ciclo euleriano, determine se o grafo tem um caminho euleriano e construa um caminho desses, se existir.

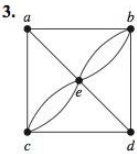
1.



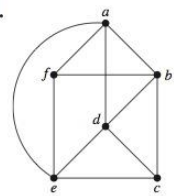
2.



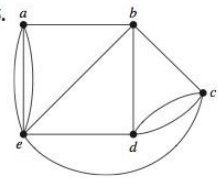
3.



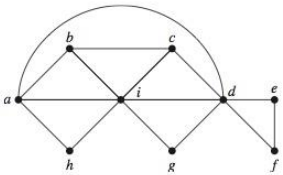
4.



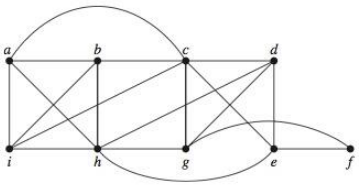
5.



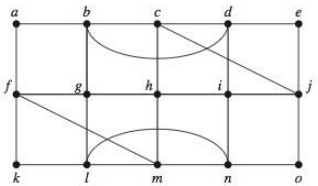
6.



7.

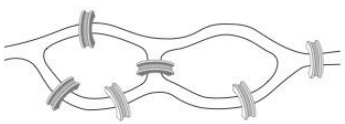


8.



9. Em Kaliningrad (o nome russo para Königsberg), existem duas pontes adicionais além das sete que estavam presentes no século XVIII. Estas novas pontes conectam as regiões *B* e *C* e as regiões *B* e *D*. Alguém pode cruzar todas as nove pontes em Kaliningrad exatamente uma vez e voltar ao ponto inicial?

10. Alguém pode cruzar todas as pontes mostradas neste mapa exatamente uma vez e retornar ao ponto inicial?

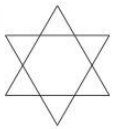


11. Quando as linhas centrais das ruas de uma cidade podem ser pintadas sem percorrer uma rua mais de uma vez? (Suponha que todas as ruas sejam de mão dupla.)

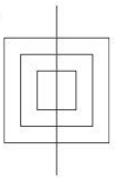
12. Descubra um procedimento, parecido com o Algoritmo 1, para construir caminhos eulerianos em multigrafos.

Nos exercícios 13 a 15, determine se a figura mostrada pode ser desenhada com um lápis em um movimento contínuo, sem erguê-lo ou retraçar parte da figura.

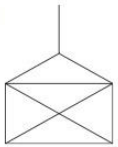
13.



14.



15.

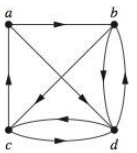


***16.** Mostre que um multigrafo orientado que não tem vértices isolados tem um ciclo euleriano se e somente se o grafo for fracamente conexo e o grau de entrada e o grau de saída de cada vértice forem iguais.

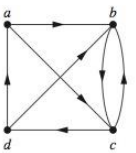
***17.** Mostre que um multigrafo orientado que não tenha vértices isolados tem um caminho euleriano, mas não um ciclo euleriano se e somente se o grafo for fracamente conexo e o grau de entrada e o grau de saída de cada vértice forem iguais para todos exceto dois vértices, um que tem o grau de entrada um a mais que seu grau de saída e outro que tem seu grau de saída um a mais que seu grau de entrada.

Nos exercícios 18 a 23, determine se o grafo orientado mostrado tem um ciclo euleriano. Construa um ciclo euleriano, se existir. Se não existir nenhum ciclo euleriano, determine se o grafo orientado tem um caminho euleriano. Construa um caminho euleriano, se existir.

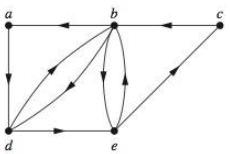
18.



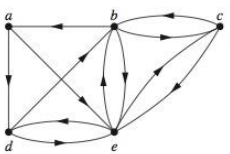
19.



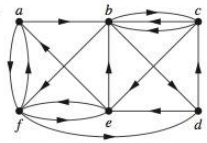
20.



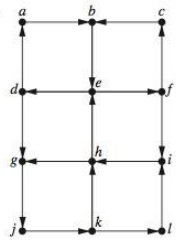
21.



22.



23.



***24.** Descubra um algoritmo para construir um ciclo euleriano em grafos orientados.

25. Descubra um algoritmo para construir caminhos eulerianos em grafos orientados.

26. Para quais valores de n estes grafos têm um ciclo euleriano?

a) K_n **b)** C_n **c)** W_n **d)** Q_n

27. Para quais valores de n os grafos do Exercício 26 tem um ciclo euleriano, mas não um caminho euleriano?

28. Para quais valores de m e n o grafo bipartido completo $K_{m,n}$ tem um

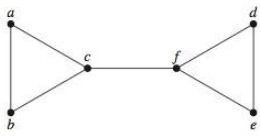
a) ciclo euleriano?

b) caminho euleriano?

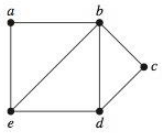
29. Encontre o menor número de vezes que é necessário erguer um lápis do papel para desenhar cada um dos grafos dos exercícios 1 a 7 sem retraçar qualquer parte do grafo.

Nos exercícios 30 a 36, determine se o grafo dado tem um ciclo hamiltoniano. Se tiver, encontre tal ciclo. Se não tiver, dê um argumento para mostrar por que não existe tal ciclo.

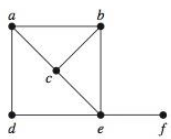
30.



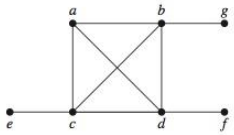
31.



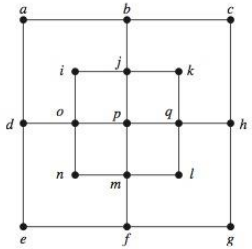
32.



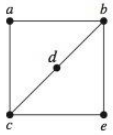
33.



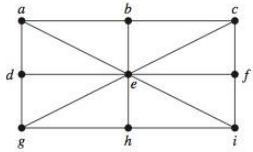
34.



35.



36.



37. O grafo do Exercício 30 tem um caminho hamiltoniano? Se tiver, encontre tal caminho. Senão, dê um argumento para mostrar por que não existe esse caminho.

38. O grafo do Exercício 31 tem um caminho hamiltoniano? Se tiver, encontre esse caminho. Senão, dê um argumento para mostrar por que não existe esse caminho.

39. O grafo do Exercício 32 tem um caminho hamiltoniano? Se tiver, encontre esse caminho. Senão, dê um argumento para mostrar por que não existe esse caminho.

40. O grafo do Exercício 33 tem um caminho hamiltoniano? Se tiver, encontre esse caminho. Senão, dê um argumento para mostrar por que não existe esse caminho.

***41.** O grafo do Exercício 34 tem um caminho hamiltoniano? Se tiver, encontre esse caminho. Senão, dê um argumento para mostrar por que não existe esse caminho.

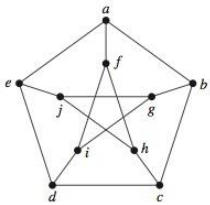
42. O grafo do Exercício 35 tem um caminho hamiltoniano? Se tiver, encontre esse caminho. Senão, dê um argumento para mostrar por que não existe esse caminho.

43. O grafo do Exercício 36 tem um caminho hamiltoniano? Se tiver, encontre esse caminho. Senão, dê um argumento para mostrar por que não existe esse caminho.

44. Para quais valores de n os grafos do Exercício 26 tem um ciclo hamiltoniano?

45. Para quais valores de m e n o grafo bipartido completo $K_{m,n}$ tem um ciclo hamiltoniano?

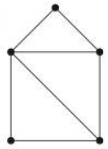
***46.** Mostre que o **grafo de Petersen**, mostrado aqui, não tem um ciclo hamiltoniano, mas que o subgrafo obtido excluindo um vértice v e todas as arestas incidentes em v tem um ciclo hamiltoniano.



47. Para cada um destes grafos, determine (*i*) se o Teorema de Dirac pode ser usado para mostrar que o grafo tem um ciclo hamiltoniano, (*ii*) se o Teorema de Ore pode ser usado para mostrar que o grafo tem um ciclo hamiltoniano e (*iii*) se o grafo tem um ciclo hamiltoniano.

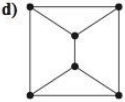
a)

b)



c)

d)



48. Você pode encontrar um grafo simples com n vértices com $n \geq 3$ que não tenha um ciclo hamiltoniano, mas que o grau de cada vértice no grafo seja, pelo menos, $(n-1)/2$?

***49.** Mostre que existe um código de Gray de ordem n sempre que n for um inteiro positivo ou, equivalentemente, mostre que o n-cubo Q_n, $n > 1$, sempre tem um ciclo hamiltoniano. [*Dica*: Use indução matemática. Mostre como produzir um código de Gray de ordem n a partir de um de ordem $n-1$.]

O **algoritmo de Fleury** para construir ciclos eulerianos começa com um vértice arbitrário de um multigrafo conexo e forma um ciclo escolhendo arestas sucessivamente. Uma vez que uma aresta tenha sido escolhida, ela é removida. As arestas são escolhidas sucessivamente de modo que cada aresta comece onde a última aresta terminou, e de modo que esta aresta não seja uma aresta de corte, a menos que não exista alternativa.

50. Use o algoritmo de Fleury para encontrar um ciclo euleriano no grafo G da Figura 5.

***51.** Expresse o algoritmo de Fleury em pseudocódigo.

****52.** Demonstre que o algoritmo de Fleury sempre produz um ciclo euleriano.

***53.** Dê uma variante do algoritmo de Fleury para produzir caminhos eulerianos.

54. Uma mensagem diagnóstica pode ser enviada em uma rede de computadores para fazer testes em todos os links e todos os dispositivos. Que tipo de caminhos deveria ser usado para testar todos os links? Para testar todos os dispositivos?

55. Mostre que um grafo bipartido com um número ímpar de vértices não tem um ciclo hamiltoniano.

Links

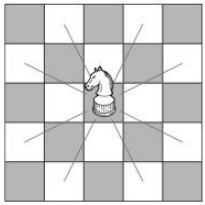
JULIUS PETER CHRISTIAN PETERSEN (1839–1910) Julius Petersen nasceu na cidade dinamarquesa de Soro. Seu pai foi um tintureiro. Em 1854, seus pais não foram mais capazes de pagar por sua educação, de modo que ele se tornou um aprendiz no armazém do seu tio. Quando seu tio morreu, deixou a Petersen dinheiro suficiente para voltar à escola. Depois de se formar, ele começou a estudar engenharia na Escola Politécnica de Copenhague, decidindo mais tarde se concentrar em matemática. Publicou seu primeiro livro-texto, um livro sobre logaritmos, em 1858. Quando sua herança se esgotou, ele teve que lecionar para ganhar a vida. De 1859 a 1871, Petersen ensinou em uma escola particular de prestígio em Copenhague. Enquanto lecionava no ensino médio, ele continuou seus estudos, ingressando na Universidade de Copenhague em 1862. Ele casou com Laura Bertelsen em 1862; tiveram três crianças, dois filhos e uma filha.

Petersen se formou em matemática na Universidade de Copenhague em 1866 e finalmente obteve seu doutorado nessa escola em 1871. Depois de receber seu doutorado, ele ensinou em uma academia politécnica e militar. Em 1887, foi nomeado professor na Universidade de Copenhague. Petersen era bem conhecido na Dinamarca como autor de uma grande série de livros-texto para ensino médio e universitário. Um de seus livros, *Métodos e Teorias para a Solução de Problemas de Construção Geométrica*, foi traduzido para oito línguas, com a versão para língua inglesa impressa pela última vez em 1960 e a versão em francês reimpressa em 1990, mais de um século da data de publicação original.

Petersen trabalhou em um grande leque de áreas, incluindo álgebra, análise, criptografia, geometria, mecânica, economia matemática e teoria dos números. Suas contribuições para a teoria dos grafos, incluindo resultados sobre grafos regulares, são seus trabalhos mais bem conhecidos. Ele se destacava pela clareza de suas exposições, pelas suas habilidades em resolver problemas, originalidade, senso de humor, vigor e didática. Um fato interessante sobre Petersen foi que ele preferia não ler os escritos de outros matemáticos. Isto o levou a redescobrir com freqüência resultados que já haviam sido demonstrados por outros, em geral com conseqüências embaraçosas. Apesar disso, ele ficava freqüentemente bravo quando os outros matemáticos não liam seus trabalhos!

A morte de Petersen foi notícia de primeira página em Copenhague. Um jornal da época o descreveu como o Hans Christian Andersen da Ciência – uma criança do povo que se saiu bem no mundo acadêmico.

Um **cavalo** é uma peça de xadrez que pode se mover ou dois espaços horizontalmente e um espaço verticalmente ou um espaço horizontalmente e dois espaços verticalmente. Isto é, um cavalo no quadrado (x, y) pode se mover para qualquer um dos oito quadrados $(x \pm 2, y \pm 1)$, $(x \pm 1, y \pm 2)$ se esses quadrados estiverem no tabuleiro de xadrez, como ilustrado aqui.



Um **passeio de cavalo** é uma seqüência de movimentos legais feitos por um cavalo iniciando em algum quadrado e visitando cada quadrado exatamente uma vez. Um passeio de cavalo é chamado de **reentrante** se existir um movimento legal que leve o cavalo do último quadrado do passeio de volta para onde o passeio começou. Podemos modelar passeios de cavalos usando o grafo que tem um vértice para cada quadrado do tabuleiro, com uma aresta que conecta dois vértices se um cavalo puder legalmente se mover entre os quadrados representados por esses vértices.

56. Desenhe o grafo que representa os movimentos legais de um cavalo em um tabuleiro 3×3.

57. Desenhe o grafo que representa os movimentos legais de um cavalo em um tabuleiro 3×4.

58. **a)** Mostre que achar um passeio de cavalo em um tabuleiro $m \times n$ é equivalente a encontrar um caminho hamiltoniano no grafo que representa os movimentos legais de um cavalo naquele tabuleiro.

b) Mostre que encontrar um passeio de cavalo reentrante em um tabuleiro $m \times n$ é equivalente a encontrar um ciclo hamiltoniano no grafo correspondente.

***59.** Mostre que existe um passeio de cavalo em um tabuleiro 3×4.

***60.** Mostre que não existe nenhum passeio de cavalo em um tabuleiro 3×3.

***61.** Mostre que não existe nenhum passeio de cavalo em um tabuleiro 4×4.

62. Mostre que o grafo que representa os movimentos legais de um cavalo em um tabuleiro $m \times n$, sempre que m e n forem inteiros positivos, é bipartido.

63. Mostre que não existe nenhum passeio de cavalo reentrante em um tabuleiro $m \times n$ quando m e n foram ambos ímpares. [*Dica*: Use os exercícios 55, 58b e 62.]

***64.** Mostre que existe um passeio de cavalo em um tabuleiro 8×8. [*Dica:* Você pode construir um passeio de cavalo usando um método inventado por H. C. Warnsdorff em 1823: comece em qualquer quadrado e então se mova sempre para um quadrado conectado ao menor número de quadrados não usados. Embora este método possa não produzir sempre um passeio de cavalo, ele com freqüência o faz.]

65. As partes deste exercício esboçam uma demonstração do Teorema de Ore. Suponha que G seja um grafo simples com n vértices, $n \geq 3$, e $\text{gr}(x) + \text{gr}(y) \geq n$ sempre que x e y forem vértices não adjacentes em G. O Teorema de Ore afirma que, nestas condições, G tem um ciclo hamiltoniano.

a) Mostre que se G não tiver um ciclo hamiltoniano, então existe um outro grafo H com os mesmos vértices de G, que pode ser construído adicionando arestas a G de modo que a adição de uma única aresta produziria um ciclo hamiltoniano em H. [*Dica:* Adicione tantas arestas quanto possível em cada vértice sucessivo de G sem produzir um ciclo hamiltoniano.]

b) Mostre que existe um caminho hamiltoniano em H.

c) Sejam $v_1, v_2, \ldots, v_n$ um caminho hamiltoniano em H. Mostre que $\text{gr}(v_1) + \text{gr}(v_n) \geq n$ e que existem no máximo $\text{gr}(v_1)$ vértices não adjacentes a v_n (incluindo o próprio v_n).

d) Seja S o conjunto dos vértices que precedem cada vértice adjacente a v_1 no caminho hamiltoniano. Mostre que S contém $\text{gr}(v_1)$ vértices e $v_n \notin S$.

e) Mostre que S contém um vértice v_k, o qual é adjacente a v_n, o que implica que existem arestas conectando v_1 e v_{k+1} e v_k e v_n.

f) Mostre que a parte (e) implica que $v_1, v_2, \ldots, v_{k-1}, v_k, v_n, v_{n-1}, \ldots, v_{k+1}, v_1$ é um ciclo hamiltoniano em G. Conclua desta contradição que o Teorema de Ore é válido.

9.6 Problemas de Caminho mais Curto

Introdução

Muitos problemas podem ser modelados usando grafos com pesos associados a suas arestas. Como ilustração, considere como um sistema de aviação pode ser modelado. Construímos o modelo de grafo básico representando as cidades por vértices e os vôos por arestas. Problemas que envolvam distâncias podem ser modelados associando as distâncias entre as cidades às arestas. Problemas que envolvem horários dos vôos podem ser modelados associando o horário de partida às arestas. Problemas que envolvem tarifas podem ser modelados associando tarifas às arestas. A Figura 1 exibe três associações de pesos diferentes às arestas de um grafo que representam distâncias, horário de partida e tarifas, respectivamente.

Os grafos que têm um número associado a cada aresta são chamados de **grafos com pesos**. Grafos com pesos são usados para modelar redes de computadores. Os custos de comunicação

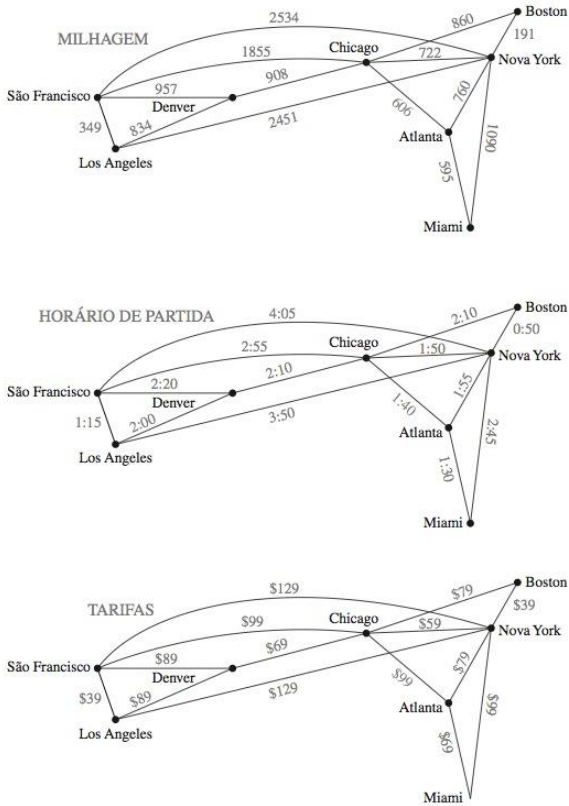


FIGURA 1 Grafos com Peso que Modelam um Sistema de Aviação.

(tal como o custo mensal de alugar uma linha telefônica), os tempos de resposta dos computadores nestas linhas ou a distância entre os computadores, podem todos ser estudados usando grafos com pesos. A Figura 2 mostra grafos com pesos que representam três maneiras de associar pesos às arestas de um grafo de uma rede de computadores, que correspondem à distância, ao tempo de resposta e ao custo.

Diversos tipos de problemas que envolvem grafos com pesos aparecem com freqüência. Determinar o caminho de menor comprimento entre dois vértices em uma rede é um desses problemas. Para ser mais específico, considere o **comprimento** de um caminho em um grafo com peso como a soma dos pesos das arestas deste caminho. (O leitor deve observar que este uso do termo *comprimento* é diferente do uso de *comprimento* para indicar o número de arestas em um caminho em um grafo sem pesos.) A questão é: qual é o caminho mais curto, isto é, o caminho de menor comprimento, entre dois vértices dados? Por exemplo, no sistema de aviação representado pelo grafo com peso mostrado na Figura 1, qual é o caminho mais curto em distância aérea

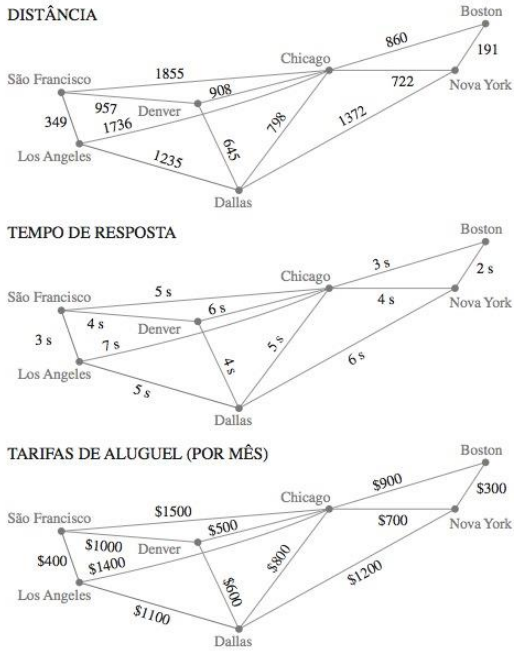


FIGURA 2 Grafos com Pesos que Modelam uma Rede de Computadores.

entre Boston e Los Angeles? Que combinações de vôos têm o menor tempo de vôo total (isto é, o tempo total no ar, sem incluir o tempo entre os vôos) entre Boston e Los Angeles? Qual é a tarifa mais barata entre estas duas cidades? Na rede de computadores mostrada na Figura 2, qual o conjunto mais barato de linhas telefônicas necessário para conectar os computadores em São Francisco com aqueles em Nova York? Qual conjunto de linhas telefônicas dá um tempo de resposta mais rápido para comunicações entre São Francisco e Nova York? Qual conjunto de linhas tem uma distância total menor?

Um outro problema importante que envolve grafos com pesos pede um ciclo de menor comprimento total que visite todo vértice de um grafo completo exatamente uma vez. Este é o famoso *problema do caixeiro-viajante*, que pede por uma ordem na qual o caixeiro-viajante visite cada cidade em sua rota exatamente uma vez, de modo que ele percorra uma distância total mínima. Discutiremos o problema do caixeiro-viajante mais adiante nesta seção.

Um Algoritmo de Caminho mais Curto

Links

Existem diversos algoritmos diferentes para encontrar um caminho mais curto entre dois vértices em um grafo com pesos. Apresentaremos um algoritmo descoberto pelo matemático holandês Edsger Dijkstra, em 1959. A versão que descreveremos resolve este problema em um grafo com peso não orientado no qual todos os pesos são positivos. É fácil adaptá-lo para resolver problemas de caminho mais curto em grafos orientados.

Antes de dar uma apresentação formal do algoritmo, daremos um exemplo motivador.

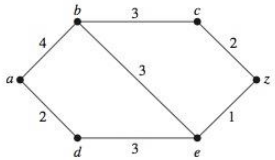


FIGURA 3 Um Grafo Simples com Peso.

EXEMPLO 1 Qual é o comprimento do caminho mais curto entre a e z no grafo com peso mostrado na Figura 3?

Solução: Embora o caminho mais curto seja facilmente encontrado por inspeção, desenvolveremos algumas idéias úteis para entender o algoritmo de Dijkstra. Resolveremos este problema encontrando o comprimento de um caminho mais curto de a a vértices sucessivos, até que z seja atingido.

Os únicos caminhos que começam em a que não contêm nenhum vértice além de a (até que o vértice final seja atingido) são a, b e a, d. Como os comprimentos de a, b e a, d são 4 e 2, respectivamente, segue que d é o vértice mais próximo de a.

Podemos encontrar o vértice mais próximo seguinte olhando todos os caminhos que passam apenas por a e d (até que o vértice final seja atingido). O mais curto destes caminhos até b ainda é a, b com comprimento 4, e o mais curto de tais caminhos até e é a, d, e, com comprimento 5. Conseqüentemente, o vértice seguinte mais próximo a a é b.

Para encontrar o terceiro vértice mais próximo a a, precisamos examinar apenas os caminhos que passam só por a, d e b (até que o vértice final seja atingido). Existe um caminho de comprimento 7 até c, a saber, a, b, c, e um caminho de comprimento 6 até z, a saber, a, d, e, z. Conseqüentemente, z é o vértice seguinte mais próximo a a, e o comprimento do caminho mais curto até z é 6. ◄

O Exemplo 1 ilustra os princípios gerais usados no algoritmo de Dijkstra. Observe que um caminho mais curto de a a z poderia ser encontrado por inspeção, entretanto, a inspeção não é prática nem para humanos nem para computadores em grafos com um grande número de arestas.

Consideraremos o problema geral de encontrar o comprimento de um caminho mais curto entre a e z em um grafo com peso simples, conexo, não orientado. O algoritmo de Dijkstra inicialmente encontra o comprimento de um caminho mais curto de a a um primeiro vértice, o comprimento mais curto de a a um segundo vértice, e assim por diante, até que o comprimento de um caminho mais curto de a a z seja encontrado.

O algoritmo se apóia em uma série de iterações. É construído um conjunto de vértices especial adicionando um vértice em cada iteração. Um procedimento de rotulação é executado em cada iteração. Neste procedimento de rotulação, um vértice w é rotulado com o comprimento de um caminho mais curto de a a w que contenha apenas vértices que já estão no conjunto especial. O vértice adicionado ao conjunto especial é o que tem rótulo mínimo entre os vértices que ainda não estão no conjunto.

Links

EDSGER WYBE DIJKSTRA (1930–2002) Edsger Dijkstra, nascido na Holanda, começou a programar computadores no início da década de 1950, quando estudava física teórica na Universidade de Leiden. Em 1952, percebendo que estava mais interessado em programação do que em física, ele completou rapidamente as exigências para o seu diploma em física e começou sua carreira como programador, mesmo que programação não fosse uma profissão reconhecida. (Em 1957, as autoridades em Amsterdam se recusaram a aceitar "programação" como profissão em sua licença de casamento. Entretanto, eles aceitaram "físico teórico" quando ele mudou aquele dado para este.)

Dijkstra foi um dos mais enfáticos proponentes da programação como uma disciplina científica. Ele fez contribuições fundamentais para áreas de sistemas operacionais, incluindo "deadlock avoidance"; linguagens de programação, incluindo a noção de programação estruturada; e algoritmos. Em 1972, Dijkstra recebeu o Turing Award da Association for Computing Machinery, um dos prêmios de maior prestígio em ciência da computação. Dijkstra se tornou um Burroughs Research Fellow em 1973, e em 1984 foi nomeado para uma cadeira em ciência da computação na Universidade do Texas, em Austin.

Daremos agora os detalhes do algoritmo de Dijkstra. Ele começa associando 0 a a e ∞ aos outros vértices. Usamos a indicação $L_0(a) = 0$ e $L_0(v) = \infty$ para estes rótulos antes de ter ocorrido qualquer iteração (o subscrito 0 significa "0-ésima" iteração). Estes rótulos são os comprimentos dos caminhos mais curtos de a aos outros vértices, em que os caminhos contêm apenas o vértice a. (Como não existe nenhum caminho de a a um vértice diferente de a, ∞ é o comprimento do caminho mais curto entre a e este vértice).

O algoritmo de Dijkstra prossegue formando um conjunto especial de vértices. Seja S_k esse conjunto depois de k iterações do procedimento de rotulação. Começamos com $S_0 = \emptyset$. O conjunto S_k é formado a partir de S_{k-1} adicionando um vértice u que não está em S_{k-1} com o menor rótulo. Uma vez que u seja adicionado a S_k, atualizamos os rótulos de todos os vértices que não estão em S_k, de modo que $L_k(v)$, o rótulo do vértice v no estágio k, é o comprimento do caminho mais curto de a a v que contém apenas vértices em S_k (isto é, vértices que já estavam no conjunto especial juntamente com u).

Seja v um vértice que não está em S_k. Para atualizar o rótulo de v, observamos que $L_k(v)$ é o comprimento do caminho mais curto de a a v que contém apenas vértices em S_k. A atualização pode ser feita de modo eficiente quando esta observação é usada: o caminho mais curto de a a v que contém apenas elementos em S_k ou é um caminho mais curto de a a v que contém apenas elementos em S_{k-1} (isto é, os vértices especiais sem incluir u) ou é um caminho mais curto de a a u no $(k-1)$ estágio com a aresta (u, v) adicionada. Em outras palavras,

$$L_k(a, v) = \min\{L_{k-1}(a, v), L_{k-1}(a, u) + w(u, v)\}.$$

Este procedimento é iterado adicionando sucessivamente vértices ao conjunto especial até que z seja adicionado. Quando z for adicionado ao conjunto especial, seu rótulo será o comprimento de um caminho mais curto de a a z. O algoritmo de Dijkstra está dado no Algoritmo 1. Mais adiante daremos uma demonstração de que este algoritmo está correto.

ALGORITMO 1 Algoritmo de Dijkstra.

procedure *Dijkstra* (G: grafo simples conexo com peso, com
todos os pesos positivos)
{G tem vértices $a = v_0, v_1, \cdots, v_n = z$ e pesos $w(v_i, v_j)$
em que $w(v_i, v_j) = \infty$ se (v_i, v_j) não for uma aresta em G}
for i := 1 **to** n
$L(v_i) := \infty$
$L(a) := 0$
$S := \emptyset$
{os rótulos agora são inicializados de modo que o rótulo de a seja 0 e todos
os outros sejam ∞, e S seja o conjunto vazio}
while $z \notin S$
begin
$u :=$ um vértice não em S, com $L(u)$ mínimo
$S := S \cup \{u\}$
for todos os vértices v não em S
if $L(u) + w(u, v) < L(v)$ **then** $L(v) := L(u) + w(u, v)$
{isso adiciona um vértice a S com rótulo mínimo e atualiza os rótulos
dos vértices que não estão em S}
end {$L(z) =$ comprimento de um caminho mais curto de a a z}

O Exemplo 2 ilustra como o algoritmo de Dijkstra funciona. Depois mostraremos que este algoritmo sempre produz o comprimento de um caminho mais curto entre dois vértices em um grafo com peso.

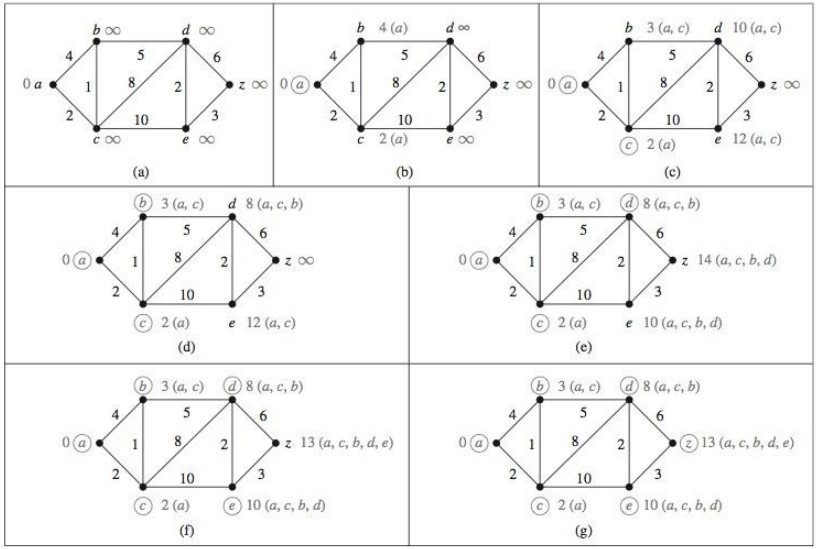


FIGURA 4 **Usando o Algoritmo de Dijkstra para Encontrar um Caminho Mais Curto de *a* a *z*.**

EXEMPLO 2 Use o algoritmo de Dijkstra para encontrar o comprimento de um caminho mais curto entre os vértices a e z no grafo com peso mostrado na Figura 4(a).

Solução: Os passos usados pelo algoritmo de Dijkstra para encontrar um caminho mais curto entre a e z estão mostrado na Figura 4. A cada iteração do algoritmo, os vértices do conjunto S_k são circulados. Um caminho mais curto de a a cada vértice que contenha apenas vértices em S_k está indicado em cada iteração. O algoritmo termina quando z for circulado. Encontramos que um caminho mais curto de a a z é a, c, b, d, e, z, com comprimento 13. ◀

Lembre-se: Ao executar o algoritmo de Dijkstra, às vezes é mais conveniente manter um registro dos rótulos dos vértices em cada passo usando uma tabela em vez de redesenhar o grafo em cada passo.

A seguir, usaremos um argumento indutivo para mostrar que o algoritmo de Dijkstra produz o comprimento de um caminho mais curto entre dois vértices a e z de um grafo com peso, conexo, não orientado. Considere como hipótese de indução a seguinte afirmação: na k-ésima iteração

(*i*) o rótulo de cada vértice v em S é o comprimento de um caminho mais curto de a a este vértice, e

(*ii*) o rótulo de cada vértice que não está em S é o comprimento de um caminho mais curto de a a este vértice que contenha apenas vértices em S (além do próprio vértice).

Quando $k = 0$, antes de qualquer iteração ser executada, $S = \emptyset$, de modo que o comprimento do caminho mais curto de a a um vértice diferente de a é ∞. Portanto, o caso base é verdadeiro.

Suponha que a hipótese de indução seja válida para a k-ésima iteração. Seja v o vértice adicionado a S na $(k + 1)$-ésima iteração, de modo que v seja um vértice que não está em S no final da k-ésima iteração com menor rótulo (em caso de empate, qualquer vértice com rótulo menor pode ser usado).

A partir da hipótese de indução, vemos que os vértices em S antes da $(k+1)$-ésima iteração são rotulados com o comprimento do caminho mais curto a partir de a. Além disso, v deve ser rotulado com o comprimento de um caminho mais curto até ele a partir de a. Se este não fosse o caso, no final da k-ésima iteração existiria um caminho de comprimento menor do que $L_k(v)$ com um vértice que não está em S [pois $L_k(v)$ é o comprimento de um caminho mais curto de a a v que contém apenas vértices em S depois da k-ésima iteração]. Seja u o primeiro vértice em tal caminho que não esteja em S. Existe um caminho de comprimento menor do que $L_k(v)$ de a a u que contém apenas vértices em S. Isto contradiz a escolha de v. Portanto, (i) vale no final da $(k+1)$-ésima iteração.

Seja u um vértice que não esteja em S depois de $k+1$ iterações. Um caminho mais curto de a a u que contenha apenas elementos em S ou contém v ou não o contém. Se ele não contiver v, então, pela hipótese de indução, seu comprimento é $L_k(u)$. Se ele contiver v, então ele é formado por um caminho de a a v de menor comprimento possível que contenha elementos de S diferentes de v, seguido pela aresta de v a u. Neste caso, seu comprimento seria $L_k(v) + w(v, u)$. Isto mostra que (ii) é verdadeira, pois $L_{k+1}(u) = \min\{L_k(u), L_k(v) + w(v, u)\}$.

Isto demonstra o Teorema 1.

TEOREMA 1 O algoritmo de Dijkstra encontra o comprimento de um caminho mais curto entre dois vértices em um grafo com peso, não orientado, simples, conexo.

Podemos agora obter uma estimativa da complexidade computacional do algoritmo de Dijkstra (em termos de adições e comparações). O algoritmo não usa mais de $n-1$ iterações, pois é adicionado um vértice ao conjunto especial em cada iteração. Teremos terminado se pudermos estimar o número de operações usadas em cada iteração. Podemos identificar o vértice que não está em S_k com o menor rótulo usando no máximo $n-1$ comparações. A seguir, usamos uma adição e uma comparação para atualizar o rótulo de cada vértice que não está em S_k. Concluímos que não mais de $2(n-1)$ operações são usadas em cada iteração, pois existem no máximo $n-1$ rótulos a serem atualizados em cada iteração. Como não usamos mais do que $n-1$ iterações, cada uma usando não mais de $2(n-1)$ operações, temos o Teorema 2.

TEOREMA 2 O algoritmo de Dijkstra usa $O(n^2)$ operações (adições e comparações) para encontrar o comprimento do caminho mais curto entre dois vértices de um grafo com peso, não orientado, simples, conexo, com n vértices.

O Problema do Caixeiro-Viajante

Discutiremos agora um problema importante que envolve grafos com pesos. Considere o seguinte problema: um caixeiro-viajante quer visitar cada uma de n cidades exatamente uma vez e voltar a seu ponto de partida. Por exemplo, suponha que o caixeiro-viajante queira visitar Detroit, Toledo, Saginaw, Grand Rapids e Kalamazoo (veja a Figura 5). Em que ordem ele deveria visitar estas cidades para percorrer a distância total mínima? Para resolver este problema, podemos supor que o caixeiro-viajante começa em Detroit (pois esta deve ser uma parte do ciclo) e examinar todas as maneiras possíveis de ele visitar as outras quatro cidades e então voltar a Detroit (começar em qualquer outro lugar produzirá os mesmos ciclos). Existe um total de 24 destes ciclos, mas porque percorremos a mesma distância quando percorremos um ciclo na ordem inversa, precisamos considerar apenas 12 ciclos diferentes para encontrar a distância mínima total que deve ser percorrida. Listamos estes 12 ciclos diferentes e a distância total percorrida em cada ciclo. Como pode ser visto a partir desta lista, a distância total mínima de 458 milhas é percorrida usando o ciclo Detroit-Toledo-Kalamazoo-Grand Rapids-Saginaw-Detroit (ou seu reverso).

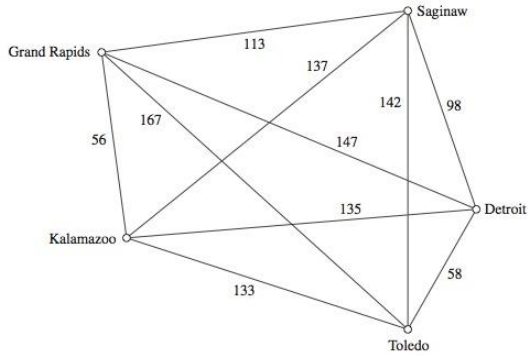


FIGURA 5 O Grafo que Mostra as Distâncias entre Cinco Cidades.

Rota	*Distância Total (milhas)*
Detroit–Toledo–Grand Rapids–Saginaw–Kalamazoo–Detroit	610
Detroit–Toledo–Grand Rapids–Kalamazoo–Saginaw–Detroit	516
Detroit–Toledo–Kalamazoo–Saginaw–Grand Rapids–Detroit	588
Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit	458
Detroit–Toledo–Saginaw–Kalamazoo–Grand Rapids–Detroit	540
Detroit–Toledo–Saginaw–Grand Rapids–Kalamazoo–Detroit	504
Detroit–Saginaw–Toledo–Grand Rapids–Kalamazoo–Detroit	598
Detroit–Saginaw–Toledo–Kalamazoo–Grand Rapids–Detroit	576
Detroit–Saginaw–Kalamazoo–Toledo–Grand Rapids–Detroit	682
Detroit–Saginaw–Grand Rapids–Toledo–Kalamazoo–Detroit	646
Detroit–Grand Rapids–Saginaw–Toledo–Kalamazoo–Detroit	670
Detroit–Grand Rapids–Toledo–Saginaw–Kalamazoo–Detroit	728

Descrevemos justamente um exemplo do **problema do caixeiro-viajante**. O problema do caixeiro-viajante pede o ciclo de peso total mínimo em um grafo não orientado, completo, com pesos, que visita cada vértice exatamente uma vez e volta a seu ponto de partida. Isto é equivalente a pedir um ciclo hamiltoniano com peso total mínimo em um grafo completo, pois cada vértice é visitado exatamente uma vez no ciclo.

A maneira mais simples de resolver um caso do problema do caixeiro-viajante é examinar todos os possíveis ciclos hamiltonianos e escolher um de comprimento total mínimo. Quantos ciclos teremos de examinar para resolver este problema se existirem n vértices no grafo? Uma vez que o ponto inicial seja escolhido, existem $(n-1)!$ ciclos hamiltonianos diferentes a serem examinados, pois existem $n-1$ escolhas para o segundo vértice, $n-2$ escolhas para o terceiro vértice e assim por diante. Como um ciclo hamiltoniano pode ser percorrido na ordem inversa, precisamos examinar apenas $(n-1)!/2$ ciclos para encontrar nossa resposta. Observe que $(n-1)!/2$ cresce extremamente rápido. Tentar resolver um problema do caixeiro-viajante desta maneira quando existem apenas poucas dúzias de vértices não é prático. Por exemplo, com 25 vértices, um total de $24!/2$ (aproximadamente $3{,}1 \times 10^{23}$) ciclos hamiltonianos diferentes deveriam ser considerados. Se levasse apenas um nanossegundo (10^{-9} segundo) para examinar cada ciclo hamiltoniano, seria necessário um total de aproximadamente 10 milhões de anos para encontrar um ciclo hamiltoniano de comprimento mínimo neste gráfico por técnicas de buscas exaustivas.

Como o problema do caixeiro-viajante tem tanto aplicações práticas quanto teóricas, uma grande quantidade de esforço tem sido dedicada a descobrir algoritmos eficientes que o resolva. Entretanto, nenhum algoritmo com complexidade com tempo polinomial no pior caso é conhecido para resolver este problema. Além disso, se um algoritmo com complexidade no tempo polinomial no pior caso fosse descoberto para o problema do caixeiro-viajante, muitos outros problemas difíceis também seriam solúveis usando algoritmos com complexidade no tempo polinomial no pior caso (tal como determinar se uma proposição em n variáveis é uma tautologia, discutido no Capítulo 1). Isto é uma conseqüência da teoria de completude NP. (Para mais informações sobre isso, consulte [GaJo79].)

Uma abordagem prática para o problema do caixeiro-viajante quando existem muitos vértices a serem visitados, é usar um **algoritmo de aproximação**. Estes são algoritmos que não produzem necessariamente a solução exata do problema, mas, em vez disso, produzem garantidamente uma solução que está próxima de uma solução exata, ou seja, eles podem produzir um ciclo hamiltoniano com peso total W' tal que $W \leq W' \leq cW$, em que W é o comprimento total de uma solução exata e c é uma constante. Por exemplo, existe um algoritmo com complexidade no tempo polinomial no pior caso que funciona se o grafo com peso satisfizer a desigualdade triangular de modo que $c = 3/2$. Para grafos com pesos, em geral, para todo número real positivo k, não é conhecido nenhum algoritmo que produza sempre uma solução no máximo k vezes a melhor solução. Se tal algoritmo existisse, isso mostraria que a classe P seria o mesmo que a classe NP, talvez a mais famosa questão em aberto sobre a complexidade de algoritmos (veja a Sessão 3.3).

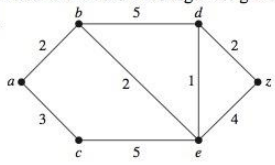
Na prática, foram desenvolvidos algoritmos que podem resolver o problema do caixeiro-viajante com até 1 000 vértices, dentro de 2% de uma solução exata, usando apenas poucos minutos de tempo de computação. Para mais informações sobre o problema do caixeiro-viajante, incluindo a história, aplicações e algoritmos, veja o capítulo sobre este tópico em *Aplicattions of Discrete Mathematics* [MiRo91], também disponível no website deste livro.

Exercícios

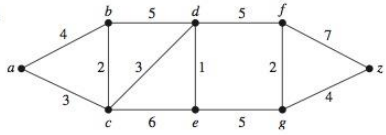
1. Para cada um destes problemas sobre um sistema de metrô, descreva um modelo de grafo com pesos que possa ser usado para resolver o problema.
- **a)** Qual a menor quantidade de tempo necessário para viajar entre duas paradas?
- **b)** Qual é a distância mínima que pode ser percorrida para atingir uma parada a partir de uma outra parada?
- **c)** Qual é a tarifa mínima para viajar entre duas paradas se as tarifas entre as paradas forem somadas para dar a tarifa total?

Nos exercícios 2 a 4, determine o comprimento de um caminho mais curto entre a e z nos seguintes grafos com pesos.

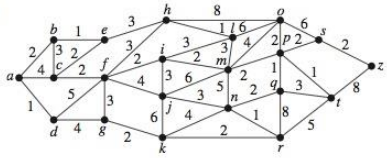
2.



3.



4.



5. Encontre o caminho mais curto entre a e z em cada um dos grafos com pesos dos exercícios 2 a 4.

6. Determine o comprimento de um caminho mais curto entre estes pares de vértices no grafo com peso do Exercício 3.
- **a)** a e d
- **b)** a e f
- **c)** c e f
- **d)** b e z

7. Encontre os caminhos mais curtos no grafo com pesos do Exercício 3 entre os pares de vértices do Exercício 6.

8. Encontre um caminho mais curto (em milhagem) entre cada um dos seguintes pares de cidades no sistema de aviação mostrado na Figura 1.
- **a)** Nova Iorque e Los Angeles
- **b)** Boston e São Francisco
- **c)** Miami e Denver
- **d)** Miami e Los Angeles

9. Encontre uma combinação de vôos com o tempo de vôo total menor entre os pares de cidades do Exercício 8, usando os tempos de vôo mostrados na Figura 1.

10. Encontre a combinação mais barata de vôos que conectam os pares de cidades do Exercício 8, usando as tarifas mostradas na Figura 1.

11. Encontre uma rota mais curta (em distância) entre os centros de computação em cada um destes pares de cidades na rede de comunicações mostrada na Figura 2.
- **a)** Boston e Los Angeles
- **b)** Nova Iorque e São Francisco
- **c)** Dalas e São Francisco
- **d)** Denver e Nova Iorque

12. Encontre uma rota com o tempo de resposta mais curto entre os pares de centro de computação do Exercício 11, usando os tempos de resposta dados na Figura 2.

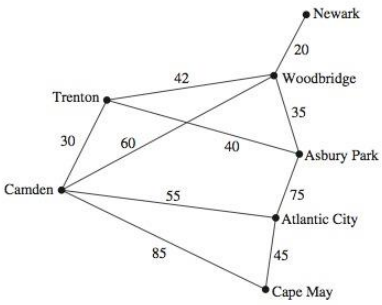
13. Encontre a rota mais barata, em tarifas de aluguel mensais, entre os pares dos centros de computação do Exercício 11 usando as tarifas de aluguel dadas na Figura 2.

14. Explique como encontrar um caminho com o menor número de arestas entre dois vértices em um grafo não orientado, considerando-o como um problema de caminho mais curto em um grafo com pesos.

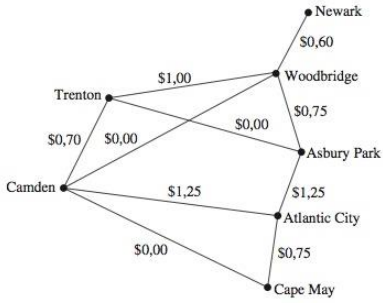
15. Estenda o algoritmo de Dijkstra para determinar o comprimento de um caminho mais curto entre dois vértices em um grafo conexo, simples, com pesos de modo que o comprimento de um caminho mais curto entre o vértice a e qualquer outro vértice do grafo seja encontrado.

16. Estenda o algoritmo de Dijkstra para determinar o comprimento de um caminho mais curto entre dois vértices em um grafo conexo, simples, com pesos de modo que um caminho mais curto entre esses vértices seja construído.

17. Os grafos com pesos nas figuras a seguir mostram algumas das estradas principais em Nova Jersey. A parte (a) mostra as distâncias entre cidades nessas estradas; a parte (b) mostra os pedágios.



(a)



(b)

- **a)** Encontre uma rota mais curta em distância entre Newark e Camden, e entre Newark e Cape May, usando essas estradas.
- **b)** Encontre uma rota mais barata em termos de pedágios totais usando as estradas do grafo entre os pares de cidades na parte (a) deste exercício.

18. O caminho mais curto entre dois vértices em um grafo com pesos é único se os pesos das arestas forem distintos?

19. Quais são algumas aplicações nas quais é necessário encontrar o comprimento de um caminho simples mais longo entre dois vértices em um grafo com pesos?

20. Qual é o comprimento de um caminho simples mais longo no grafo com pesos da Figura 4 entre a e z? E entre c e z?

O **algoritmo de Floyd**, mostrado como Algoritmo 2, pode ser usado para encontrar o comprimento de um caminho mais curto entre todos os pares de vértices em um grafo simples, conexo, com pesos. Entretanto, este algoritmo não pode ser usado para construir caminhos mais curtos. (Associamos um peso infinito a qualquer par de vértices que não esteja ligado por uma aresta do grafo).

21. Use o algoritmo de Floyd para encontrar a distância entre todos os pares de vértices no grafo com pesos da Figura 4(a).

***22.** Demonstre que o algoritmo de Floyd determina a distância mais curta entre todos os pares de vértices em um grafo simples com pesos.

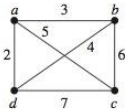
***23.** Dê uma estimativa O grande do número de operações (comparações e adições) usadas pelo algoritmo de Floyd para determinar a distância mais curta entre todos os pares de vértices em um grafo simples, com pesos, com n vértices.

***24.** Mostre que o algoritmo de Dijkstra pode não funcionar se as arestas puderem ter pesos negativos.

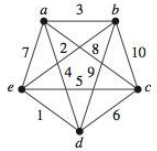
ALGORITMO 2 Algoritmo de Floyd.

procedure *Floyd*(G: grafo simples com peso)
{G tem vértices $v_1, v_2, \cdots v_n$ e pesos $w(v_i, v_j)$
com $w(v_i, v_j) = \infty$ se (v_i, v_j) não for uma aresta}
for $i := 1$ **to** n
for $j := 1$ **to** n
$d(v_i, v_j) := w(v_i, v_j)$
for $i := 1$ **to** n
for $j := 1$ **to** n
for $k := 1$ **to** n
if $d(v_j, v_i) + d(v_i, v_k) < d(v_j, v_k)$
then $d(v_j, v_k) :=$
$d(v_j, v_i) + d(v_i, v_k)$
{$d(v_i, v_j)$ é o comprimento de um caminho mais curto entre v_i e v_j}

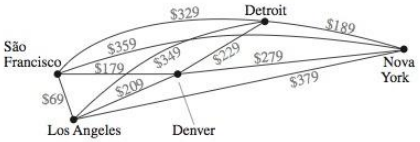
25. Resolva o problema do caixeiro-viajante para este grafo encontrando o peso total de todos os ciclos hamiltonianos e determinando um ciclo com peso total mínimo.



26. Resolva o problema do caixeiro-viajante para este grafo encontrando o peso total de todos os ciclos hamiltonianos e determinando um ciclo com peso total mínimo.



27. Encontre uma rota com a menor tarifa total que visite cada uma das cidades neste grafo, em que o peso em uma aresta é o menor preço disponível para um vôo entre as duas cidades.



28. Encontre uma rota com a menor tarifa total que visite cada uma das cidades neste grafo, em que o peso em uma aresta é o menor preço disponível para um vôo entre as duas cidades.

29. Construa um grafo não orientado com pesos, tal que o peso total de um ciclo que visita cada vértice pelo menos uma vez é minimizado para um ciclo que visita alguns dos vértices mais de uma vez. [*Dica*: Existem exemplos com três vértices.]

30. Mostre que o problema de encontrar um ciclo de peso total mínimo que visita cada vértice de um grafo com pesos pelo menos uma vez pode ser reduzido ao problema de encontrar um ciclo de peso total mínimo que visita cada vértice de um grafo com pesos exatamente uma vez. Faça isso construindo um novo grafo com pesos com os mesmos vértices e arestas que o grafo original, mas cujo peso das arestas que ligam os vértices u e v é igual ao peso total mínimo de um caminho de u a v no grafo original.

9.7 Grafos Planares

Introdução

Links

Considere o problema de ligar três casas a três serviços públicos, como mostrado na Figura 1. É possível ligar estas casas e os serviços públicos de modo que nenhuma das conexões se cruze? Este problema pode ser modelado usando o grafo bipartido completo $K_{3,3}$. A pergunta original pode ser refeita como: $K_{3,3}$ pode ser desenhado no plano de modo que nenhum par de arestas se cruze?

Nesta seção, estudaremos a possibilidade ou não de um grafo ser desenhado no plano sem arestas que se cruzem. Em particular, responderemos o problema das casas e dos serviços públicos.

Existem sempre muitas maneiras de representar um grafo. Quando é possível encontrar pelo menos uma forma de representar este grafo em um plano sem quaisquer arestas se cruzarem?

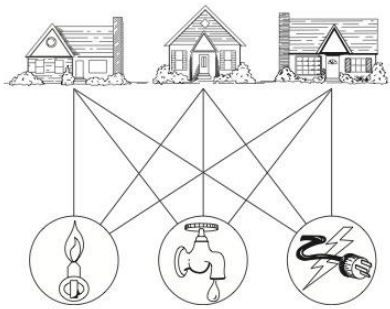


FIGURA 1 **Três Casas e Três Serviços Públicos.**

DEFINIÇÃO 1 Um grafo é dito *planar* se ele puder ser desenhado no plano sem quaisquer arestas se cruzando (em que um cruzamento de arestas é a intersecção de retas ou arcos que as representam em um ponto diferente de sua extremidade comum). Tal desenho é chamado de *representação planar* do grafo.

Um grafo pode ser planar mesmo que ele seja usualmente desenhado com cruzamentos, pois pode ser possível desenhá-lo de uma maneira diferente, sem cruzamentos.

EXEMPLO 1 K_4 (mostrado na Figura 2 com duas arestas se cruzando) é planar?

Solução: K_4 é planar porque pode ser desenhado sem cruzamentos, como mostrado na Figura 3. ◀

EXEMPLO 2 Q_3, mostrado na Figura 4, é planar?

Solução: Q_3 é planar, pois pode ser desenhado sem nenhum cruzamento de arestas, como mostrado na Figura 5. ◀

Podemos mostrar que um grafo é planar exibindo uma representação planar. É mais difícil mostrar que um grafo não é planar. Daremos um exemplo para mostrar como isto pode ser feito de uma forma *ad hoc*. Mais tarde, desenvolveremos alguns resultados gerais que podem ser usados para fazer isto.

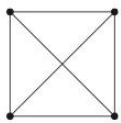


FIGURA 2 **O Grafo K_4.**

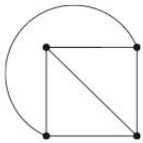


FIGURA 3 **K_4 Desenhado sem Cruzamentos.**

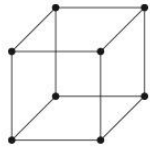


FIGURA 4 **O Grafo Q_3.**

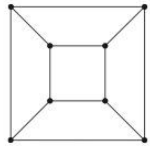


FIGURA 5 **Uma Representação Planar de Q_3.**

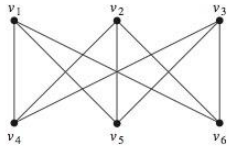


FIGURA 6 **O Grafo $K_{3,3}$.**

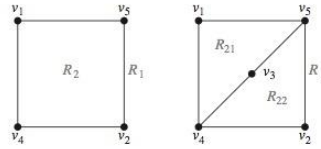


FIGURA 7 **Mostrando que $K_{3,3}$ Não É Planar.**

EXEMPLO 3 $K_{3,3}$, mostrado na Figura 6, é planar?

Solução: Qualquer tentativa de desenhar $K_{3,3}$ no plano sem cruzamento de arestas está fadada ao fracasso. Agora, mostraremos por quê. Em qualquer representação planar de $K_{3,3}$, os vértices v_1 e v_2 devem estar conectados tanto a v_4 quanto a v_5. Estes quatro vértices formam uma curva fechada que divide o plano em duas regiões, R_1 e R_2, como mostrado na Figura 7(a). O vértice v_3 ou está em R_1 ou está em R_2. Quando v_3 estiver em R_2, o interior da curva fechada, as arestas entre v_3 e v_4 e entre v_3 e v_5 separam R_2 em duas sub-regiões, R_{21} e R_{22}, como mostrado na Figura 7(b).

A seguir, observe que não existe nenhuma maneira de colocar o último vértice v_6 sem forçar um cruzamento. Pois se v_6 estiver em R_1, então a aresta entre v_6 e v_3 não pode ser desenhada sem um cruzamento. Se v_6 estiver em R_{21}, então a aresta entre v_2 e v_6 não pode ser desenhada sem um cruzamento. Se v_6 estiver em R_{22}, então a aresta entre v_1 e v_6 não pode ser desenhada sem um cruzamento.

Um argumento análogo pode ser usado quando v_3 estiver em R_1. É deixado para o leitor completar este argumento (veja o Exercício 10 no final desta seção). Segue que $K_{3,3}$ não é planar. ◀

O Exemplo 3 resolve o problema das casas e dos serviços públicos que foi descrito no início desta seção. As três casas e os três serviços públicos não podem ser conectados no plano sem um cruzamento. Um argumento análogo pode ser usado para mostrar que K_5 não é planar. (Veja o Exercício 11 no final desta seção.)

O fato de um grafo ser planar desempenha um papel importante no projeto de circuitos eletrônicos. Podemos modelar um circuito com um grafo que representa os componentes do circuito por vértices e as conexões entre eles por arestas. Podemos imprimir um circuito em uma única tábua sem nenhum cruzamento de conexões se o grafo que representa o circuito for planar. Quando este grafo não for planar, devemos nos voltar para opções mais caras. Por exemplo, podemos dividir os vértices no grafo que representa o circuito em subgrafos planares. Então, construímos o circuito original usando camadas múltiplas. (Veja o preâmbulo do Exercício 30 para aprender sobre a espessura de um grafo.) Podemos construir o circuito usando fios isolados sempre que as conexões se cruzem. Neste caso, é importante desenhar o grafo com o menor número possível de cruzamentos. (Veja o preâmbulo do Exercício 26 para aprender sobre o número de cruzamentos de um grafo.)

Fórmula de Euler

Uma representação planar de um grafo divide o plano em **regiões**, incluindo uma região ilimitada. Por exemplo, a representação planar do grafo mostrado na Figura 8 divide o plano em seis regiões. Estas estão identificadas na figura. Euler mostrou que todas as representações planares de um grafo dividem o plano no mesmo número de regiões. Ele conseguiu isto descobrindo uma relação entre o número de regiões, o número de vértices e o número de arestas de um grafo planar.

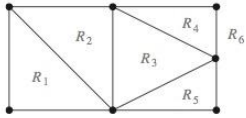


FIGURA 8 As Regiões da Representação Planar de um Grafo.

TEOREMA 1 **FÓRMULA DE EULER** Seja G um grafo simples planar conexo com e arestas e v vértices. Seja r o número de regiões em uma representação planar de G. Então, $r = e - v + 2$.

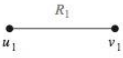


FIGURA 9 O Caso Base da Demonstração da Fórmula de Euler.

Demonstração: Inicialmente, especificamos uma representação planar de G. Demonstraremos o teorema construindo uma seqüência de subgrafos $G_1, G_2, \ldots, G_e = G$, adicionando sucessivamente uma aresta em cada estágio. Isto é feito usando a seguinte definição indutiva. Escolha arbitrariamente uma aresta de G para obter G_1. Obtenha G_n de G_{n-1} adicionando arbitrariamente uma aresta que seja incidente em um vértice já em G_{n-1}, adicionando o outro vértice incidente nesta aresta se ele ainda não estiver em G_{n-1}. Esta construção é possível porque G é conexo. G é obtido depois de e arestas terem sido adicionadas. Seja r_n, e_n e v_n o número de regiões, arestas e vértices da representação planar de G_n induzida pela representação planar de G, respectivamente.

A demonstração agora prosseguirá por indução. A relação $r_1 = e_1 - v_1 + 2$ é verdadeira para G_1, pois $e_1 = 1$, $v_1 = 2$ e $r_1 = 1$. Isto está mostrado na Figura 9.

Suponha agora que $r_n = e_n - v_n + 2$. Seja $\{a_{n+1}, b_{n+1}\}$ a aresta que é adicionada a G_n para obter-se G_{n+1}. Existem duas possibilidades a serem consideradas. No primeiro caso, tanto a_{n+1} quanto b_{n+1} já estão em G_n. Estes dois vértices devem estar na fronteira de uma região comum R, ou seria impossível adicionar a aresta $\{a_{n+1}, b_{n+1}\}$ a G_n sem que duas arestas se cruzassem (e G_{n+1} é planar). A adição desta nova aresta divide R em duas regiões. Conseqüentemente, neste caso, $r_{n+1} = r_n + 1$, $e_{n+1}, = e_n + 1$ e $v_{n+1} = v_n$. Assim, cada lado da fórmula que relaciona o número de regiões, de arestas e de vértices aumenta exatamente um, de modo que a fórmula ainda é verdadeira. Em outras palavras, $r_{n+1} = e_{n+1} - v_{n+1} + 2$. Este caso está ilustrado na Figura 10(a).

No segundo caso, um dos dois vértices da nova aresta ainda não está em G_n. Suponha que a_{n+1} esteja em G_n, mas que b_{n+1} não. A adição desta nova aresta não produz nenhuma região nova, pois b_{n+1} deve estar em uma região que tem a_{n+1} em sua fronteira. Conseqüentemente, $r_{n+1} = r_n$. Além disso, $e_{n+1} = e_n + 1$ e $v_{n+1} = v_n + 1$. Cada lado da fórmula que relaciona o número

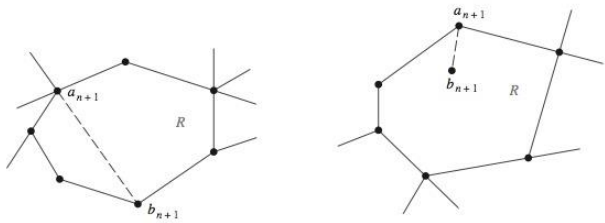


FIGURA 10 Adicionando uma Aresta a G_n para Produzir G_{n+1}.

de regiões, de arestas e de vértices permanece o mesmo, de modo que a fórmula ainda é verdadeira. Em outras palavras, $r_{n+1} = e_{n+1} - v_{n+1} + 2$. Este caso está ilustrado na Figura 10(b).

Completamos o argumento de indução. Portanto, $r_n = e_n - v_n + 2$ para todo n. Como o grafo original é o grafo G_e, obtido depois de e arestas terem sido adicionadas, o teorema é verdadeiro. ◁

A fórmula de Euler está ilustrada no Exemplo 4.

EXEMPLO 4 Suponha que um grafo simples planar conexo tenha 20 vértices, cada um de grau 3. Em quantas regiões uma representação deste grafo planar divide o plano?

Solução: Este grafo tem 20 arestas, cada uma de grau 3, de modo que $v = 20$. Como a soma dos graus dos vértices, $3v = 3 \cdot 20 = 60$, é igual a duas vezes o número de arestas, $2e$, temos $2e = 60$, ou $e = 30$. Conseqüentemente, a partir da fórmula de Euler, o número de regiões é

$$r = e - v + 2 = 30 - 20 + 2 = 12.$$ ◀

A fórmula de Euler pode ser usada para estabelecer algumas desigualdades que devem ser satisfeitas pelos grafos planares. Uma de tais desigualdades é dada no Corolário 1.

COROLÁRIO 1 Se G é um grafo simples planar conexo com e arestas e v vértices, em que $v \geq 3$, então $e \leq 3v - 6$.

Antes de demonstrar o Corolário 1, o usaremos para demonstrar o seguinte resultado útil.

COROLÁRIO 2 Se G é um grafo simples planar conexo, então G tem um vértice de grau que não excede 5.

Demonstração: Se G tiver um ou dois vértices, o resultado é verdadeiro. Se G tiver pelo menos três vértices, pelo Corolário 1 sabemos que $e \leq 3v - 6$, de modo que $2e \leq 6v - 12$. Se o grau de cada vértice fosse pelo menos seis, então, como $2e = \sum_{v \in V} \text{gr}(v)$ (pelo Teorema do Aperto de Mãos), teríamos $2e \geq 6v$. Mas isto contradiz a desigualdade $2e \leq 6v - 12$. Segue que deve existir pelo menos um vértice com grau que não seja maior do que cinco. ◁

A demonstração do Corolário 1 é baseada no conceito do **grau** de uma região, que é definido como o número de arestas na fronteira desta região. Quando uma aresta ocorre duas vezes na fronteira (de modo que ela é percorrida duas vezes quando traçamos a fronteira), ela contribui dois para o grau. Os graus das regiões do grafo mostrado na Figura 11 estão indicados na figura.

Agora a demonstração do Corolário 1 pode ser dada.

Demonstração: Um grafo simples planar conexo desenhado no plano divide o plano em regiões, digamos r delas. O grau de cada região é, pelo menos, três. (Pois os grafos discutidos aqui são grafos simples, nos quais não são permitidas arestas múltiplas, que poderiam produzir regiões de grau dois, ou laços, que poderiam produzir regiões de grau um.) Em particular, observe que o grau da região ilimitada é, pelo menos, três, pois existem, pelo menos, três vértices no grafo.

Observe que a soma dos graus das regiões é exatamente duas vezes o número de arestas no grafo, pois cada aresta ocorre na fronteira de uma região exatamente duas vezes (ou em duas

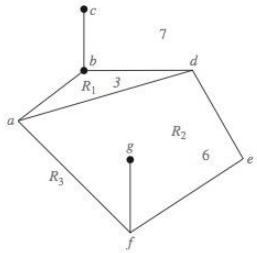


FIGURA 11 Os Graus das Regiões.

regiões diferentes ou duas vezes na mesma região). Como cada região tem grau maior que ou igual a três, segue que

$$2e = \sum_{\text{todas as regiões } R} \text{gr(R)} \geq 3r.$$

Portanto,

$$(2/3)e \geq r.$$

Usando $r = e - v + 2$ (fórmula de Euler), obtemos

$$e - v + 2 \leq (2/3)e.$$

Segue que $e/3 \leq v - 2$. Isto mostra que $e \leq 3v - 6$. ◁

Este corolário pode ser usado para mostrar que K_5 não é planar.

EXEMPLO 5 Mostre que K_5 não é planar usando o Corolário 1.

Solução: O grafo K_5 tem cinco vértices e 10 arestas. Entretanto, a desigualdade $e \leq 3v - 6$ não é satisfeita para este grafo, pois $e = 10$ e $3v - 6 = 9$. Portanto, K_5 não é planar. ◀

Foi mostrado anteriormente que $K_{3,3}$ não é planar. Observe, entretanto, que este grafo tem seis vértices e nove arestas. Isto significa que a desigualdade $e = 9 \leq 12 = 3 \cdot 6 - 6$ é satisfeita. Conseqüentemente, o fato de a desigualdade $e \leq 3v - 6$ ser satisfeita *não* implica que o gráfico seja planar. Entretanto, o seguinte corolário do Teorema 1 pode ser usado para mostrar que $K_{3,3}$ não é planar.

COROLÁRIO 3 Se um grafo simples planar conexo tem e arestas e v vértices, com $v \geq 3$, e nenhum ciclo de comprimento três, então $e \leq 2v - 4$.

A demonstração do Corolário 3 é análoga à do Corolário 1, exceto que neste caso o fato de não haver ciclos de comprimento três implica que o grau de uma região deve ser, pelo menos, quatro. Os detalhes desta demonstração são deixados para o leitor (veja o Exercício 15 no final desta seção).

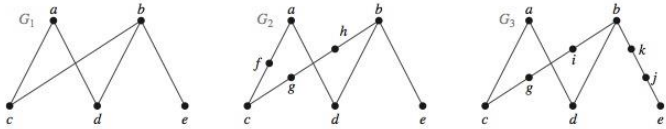


FIGURA 12 Grafos Homeomorfos.

EXEMPLO 6 Use o Corolário 3 para mostrar que $K_{3,3}$ não é planar.

Solução: Como $K_{3,3}$ não tem nenhum ciclo de comprimento três (isto é fácil de ver, pois ele é bipartido), o Corolário 3 pode ser usado. $K_{3,3}$ tem seis vértices e nove arestas. Como $e = 9$ e $2v - 4 = 8$, o Corolário 3 mostra que $K_{3,3}$ não é planar. ◀

Teorema de Kuratowski

Vimos que $K_{3,3}$ e K_5 não são planares. Claramente, um grafo não é planar se contiver um destes dois grafos como subgrafos. Surpreendentemente, todos os grafos que não são planares devem conter um subgrafo que pode ser obtido de $K_{3,3}$ ou K_5 usando certas operações permitidas.

Se um grafo for planar, qualquer grafo obtido da remoção de uma aresta $\{u, v\}$ e adição de um novo vértice w e das arestas $\{u, w\}$ e $\{w, v\}$ também o será. Tal operação é chamada de **subdivisão elementar**. Os grafos $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ são ditos **homeomorfos** se eles puderem ser obtidos do mesmo grafo por uma seqüência de subdivisões elementares.

EXEMPLO 7 Mostre que os grafos G_1, G_2 e G_3 mostrados na Figura 12 são todos homeomorfos.

Solução: Estes três grafos são homeomorfos porque todos podem ser obtidos a partir de G_1 por subdivisões elementares. G_1 pode ser obtido de si mesmo por uma seqüência vazia de subdivisões elementares. Para obter G_2 de G_1, podemos usar esta seqüência de subdivisões elementares: (*i*) remova a aresta $\{a, c\}$, adicione o vértice f e as arestas $\{a, f\}$ e $\{f, c\}$; (*ii*) remova a aresta $\{b, c\}$, adicione o vértice g e as arestas $\{b, g\}$ e $\{g, c\}$; e (*iii*) remova a aresta $\{b, g\}$, adicione o vértice h e as arestas $\{g, h\}$ e $\{b, h\}$. Deixamos para o leitor a determinação da seqüência de subdivisões elementares necessárias para obter G_3 de G_1. ◀

O matemático polonês Kazimierz Kuratowski estabeleceu o Teorema 2 em 1930, que caracteriza os grafos planares usando o conceito de homeomorfismo de grafo.

Links

KAZIMIERZ KURATOWSKI (1896–1980) Kazimierz Kuratowski, filho de um famoso advogado de Varsóvia, freqüentou o ensino médio em Varsóvia. Ele estudou em Glasgow, Escócia, de 1913 a 1914, mas não pode retornar para lá depois da eclosão da Primeira Guerra Mundial. Em 1915, ingressou na Universidade de Varsóvia, onde foi ativo no movimento estudantil patriótico polonês. Ele publicou seu primeiro artigo em 1919 e obteve seu doutorado em 1921. Foi um membro ativo do grupo conhecido como a Escola de Matemática de Varsóvia, tendo trabalhado nas áreas de fundamentos de teoria dos conjuntos e topologia. Ele foi nomeado professor associado na Lwów Polytechnical University, onde permaneceu por sete anos, colaborando com os importantes matemáticos poloneses Banach e Ulam. Em 1930, enquanto em Lwów, Kuratowski completou seu trabalho sobre a caracterização de grafos planares.

Em 1934, retornou à Universidade de Varsóvia como professor titular. Até o início da Segunda Guerra Mundial, esteve ativo em pesquisa e ensino. Durante a guerra, por causa da perseguição aos poloneses que tinham instrução, Kuratowski se escondeu sob um nome fictício e ensinou na Universidade clandestina de Varsóvia. Depois da guerra, ele ajudou a reviver a matemática polonesa, servindo como diretor do Polish National Mathematics Institute. Ele escreveu mais de 180 artigos e três livros-textos amplamente utilizados.

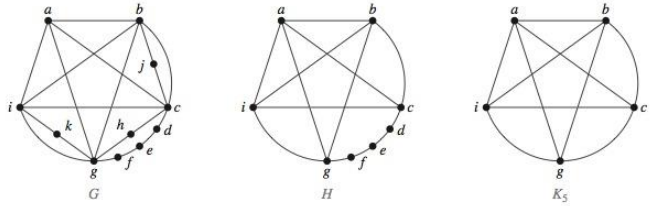


FIGURA 13 **O Grafo Não Orientado *G*, um Subgrafo *H* Homeomorfo a K_5, e K_5.**

TEOREMA 2 Um grafo não é planar se e somente se ele contiver um subgrafo homeomorfo a $K_{3,3}$ ou K_5.

É claro que um grafo que contenha um subgrafo homeomorfo a $K_{3,3}$ ou K_5 não é planar. Entretanto, a demonstração da recíproca, ou seja, que todo grafo não planar contém um subgrafo homeomorfo a $K_{3,3}$ ou K_5, é complicada e não será fornecida aqui. Os exemplos 8 e 9 ilustram como o Teorema de Kuratowski é usado.

EXEMPLO 8 Determine se o grafo *G* mostrado na Figura 13 é planar.

Exemplos Extras

Solução: *G* tem um subgrafo *H* homeomorfo a K_5. *H* é obtido excluindo *h*, *j* e *k* e todas as arestas incidentes a estes vértices. *H* é homeomorfo a K_5 porque ele pode ser obtido a partir de K_5 (com vértices *a*, *b*, *c*, *g* e *i*) por uma seqüência de subdivisões elementares, adicionando os vértices *d*, *e* e *f*. (O leitor deve construir tal seqüência de subdivisões elementares.) Portanto, *G* não é planar. ◄

EXEMPLO 9 O grafo de Petersen, mostrado na Figura 14(a), é planar? (O matemático dinamarquês Julius Petersen estudou este grafo em 1891; ele é freqüentemente usado para ilustrar várias propriedades teóricas dos grafos.)

Solução: O subgrafo *H* do grafo de Petersen, obtido excluindo *b* e as três arestas que tem *b* como uma extremidade, mostrado na Figura 14(b), é homeomorfo a $K_{3,3}$, com conjunto de

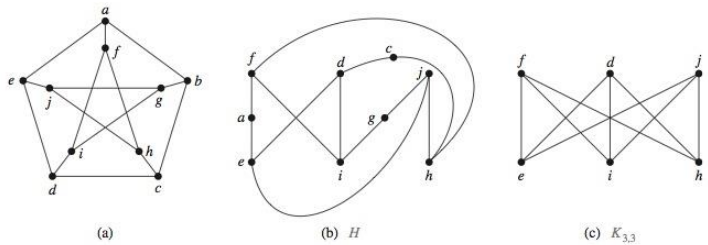


FIGURA 14 **(a) O Grafo de Petersen, (b) um Subgrafo *H* Homeomorfo a $K_{3,3}$, e (c) $K_{3,3}$.**

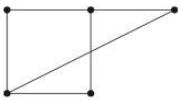
vértices $\{f, d, j\}$ e $\{e, i, h\}$, pois ele pode ser obtido por uma seqüência de subdivisões elementares, excluindo $\{d, h\}$ e adicionando $\{c, h\}$ e $\{c, d\}$, excluindo $\{e, f\}$ e adicionando $\{a, e\}$ e $\{a, f\}$, e excluindo $\{i, j\}$ e adicionando $\{g, i\}$ e $\{g, j\}$. Portanto, o grafo de Petersen não é planar. ◀

Exercícios

1. Cinco casas podem ser conectadas a dois serviços públicos sem que as conexões se cruzem?

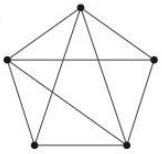
Nos exercícios 2 a 4, desenhe o grafo planar dado sem nenhum cruzamento.

2.



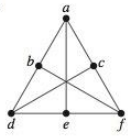
3.

4.

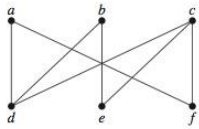


Nos exercícios 5 a 9, determine se o grafo dado é planar. Se for, desenhe-o sem cruzamento de arestas.

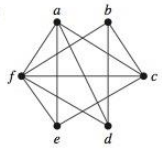
5.



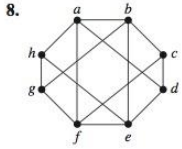
6.



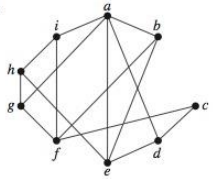
7.



8.



9.



10. Complete o argumento no Exemplo 3.

11. Mostre que K_5 não é planar usando um argumento análogo ao dado no Exemplo 3.

12. Suponha que um grafo planar conexo tenha oito vértices, cada um com grau três. Em quantas regiões uma representação planar deste grafo divide o plano?

13. Suponha que um grafo planar conexo tenha seis vértices, cada um de grau quatro. Em quantas regiões uma representação planar deste grafo divide o plano?

14. Suponha que um grafo planar conexo tenha 30 arestas. Se uma representação planar deste grafo divide o plano em 20 regiões, quantos vértices este grafo tem?

15. Demonstre o Corolário 3.

16. Suponha que um grafo simples, planar, bipartido, conexo tenha e arestas e v vértices. Mostre que $e \leq 2v - 4$ se $v \geq 3$.

***17.** Suponha que um grafo simples, planar, conexo, com e arestas e v vértices não contenha nenhum ciclo simples de comprimento menor que ou igual a 4. Mostre que $e \leq (5/3)v - (10/3)$ se $v \geq 4$.

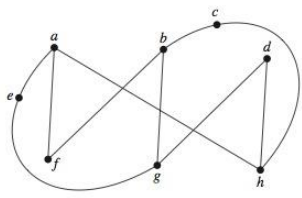
18. Suponha que um grafo planar tenha k componentes conexas, e arestas e v vértices. Suponha também que o plano seja dividido em r regiões por uma representação planar do grafo. Encontre uma fórmula para r em termos de e, v e k.

19. Quais destes grafos não planares têm a propriedade em que a remoção de qualquer vértice e de todas as arestas incidentes a esse vértice produz um grafo planar?

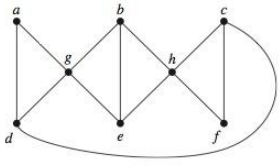
a) K_5 **b)** K_6 **c)** $K_{3,3}$ **d)** $K_{3,4}$

Nos exercícios 20 a 22, determine se o grafo dado é homeomorfo a $K_{3,3}$.

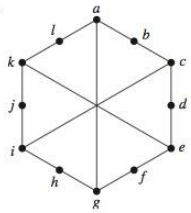
20.



21.

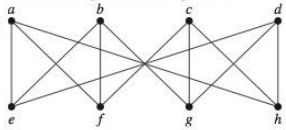


22.

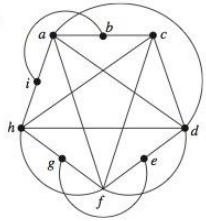


Nos exercícios 23 a 25, use o Teorema de Kuratowski para determinar se o grafo dado é planar.

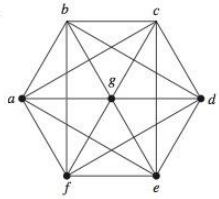
23.



24.



25.



O número de cruzamentos de um grafo simples é o número mínimo de cruzamentos que pode ocorrer quando este grafo é desenhado no plano no qual nunca é permitido a três arcos que representam arestas cruzar no mesmo ponto.

26. Mostre que o número de cruzamentos de $K_{3,3}$ é 1.

****27.** Encontre o número de cruzamentos de cada um destes grafos não planares.

a) K_5 **b)** K_6 **c)** K_7
d) $K_{3,4}$ **e)** $K_{4,4}$ **f)** $K_{5,5}$

***28.** Encontre o número de cruzamentos do grafo de Petersen.

****29.** Mostre que se m e n forem inteiros positivos pares, o número de cruzamentos de $K_{m,n}$ é menor que ou igual a $mn(m-2)(n-2)/16$. [*Dica:* Coloque m vértices ao longo do eixo x de modo que eles estejam igualmente espaçados e simétricos em relação à origem, e coloque n vértices ao longo do eixo y de modo que eles estejam igualmente espaçados e simétricos em relação à origem. Agora conecte cada um dos m vértices no eixo x a cada um dos vértices no eixo y e conte os cruzamentos.]

A **espessura** de um grafo simples G é o menor número de subgrafos planares de G que tem G como sua união.

30. Mostre que a espessura de $K_{3,3}$ é 2.

***31.** Encontre a espessura dos grafos do Exercício 27.

32. Mostre que se G for um grafo simples, conexo, com v vértices e e arestas, então a espessura de G é, pelo menos, $\lceil e/(3v-6) \rceil$.

***33.** Use o Exercício 32 para mostrar que espessura de K_n é, pelo menos, $\lfloor (n+7)/6 \rfloor$ sempre que n for um inteiro positivo.

34. Mostre que se G for um grafo simples, conexo, com v vértices e e arestas e sem nenhum ciclo de comprimento 3, então a espessura de G é, pelo menos, $\lceil e/(2v-4) \rceil$.

35. Use o Exercício 34 para mostrar que a espessura de $K_{m,n}$ é, pelo menos, $\lceil mn/(2m+2n-4) \rceil$ sempre que m e n forem inteiros positivos.

***36.** Desenhe K_5 na superfície de um toro (um sólido na forma de uma rosca) de modo que não haja cruzamento de arestas.

***37.** Desenhe $K_{3,3}$ na superfície de um toro de modo que não haja cruzamento de arestas.

9.8 Coloração de Grafos

Introdução

Links

Problemas relacionados à coloração de mapas de regiões, tais como mapas de partes do mundo, têm gerado muitos resultados na Teoria dos Grafos. Quando um mapa* é colorido, duas regiões com uma fronteira comum são associadas em geral a cores diferentes. Uma maneira de garantir que duas regiões adjacentes nunca tenham a mesma cor é usar uma cor diferente para cada região. Entretanto, isto é ineficiente, e em mapas com muitas regiões seria difícil distinguir entre cores

* Suporemos que todas as regiões em um mapa são conexas. Isto elimina quaisquer problemas causados por entidades geográficas tais como Michigan.

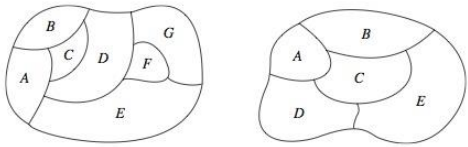


FIGURA 1 **Dois Mapas.**

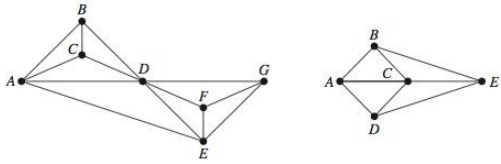


FIGURA 2 **Grafos Duais dos Mapas da Figura 1.**

parecidas. Em vez disso, um pequeno número de cores deveria ser usado sempre que possível. Considere o problema de determinar o menor número de cores que pode ser usado para colorir um mapa de modo que regiões adjacentes nunca tenham a mesma cor. Por exemplo, para o mapa mostrado à esquerda na Figura 1, quatro cores seriam suficientes, mas três cores não seriam suficientes. (O leitor deve verificar isso.) No mapa à direita na Figura 1, três cores são suficientes (mas duas não são).

Cada mapa no plano pode ser representado por um grafo. Para montar esta correspondência, cada região do mapa é representada por um vértice. Arestas conectam dois vértices se as regiões representadas por estes vértices tiverem uma fronteira em comum. Duas regiões que se toquem em apenas um ponto não são consideradas adjacentes. O grafo resultante é chamado de **grafo dual** do mapa. Pela maneira na qual os grafos duais de mapas são construídos, é claro que qualquer mapa no plano tem um grafo dual planar. A Figura 2 exibe os grafos duais que correspondem aos mapas da Figura 1.

O problema de colorir as regiões de um mapa é equivalente ao problema de colorir os vértices do grafo dual, de modo que dois vértices adjacentes não tenham a mesma cor. Definimos agora a coloração de um grafo.

DEFINIÇÃO 1 Uma *coloração* de um grafo simples é a associação de uma cor a cada vértice do grafo de modo que dois vértices adjacentes não estejam associados à mesma cor.

Um grafo pode ser colorido associando uma cor diferente a cada um de seus vértices. Entretanto, para a maioria dos grafos pode ser encontrada uma coloração que usa menos cores que o número de vértices do grafo. Qual é o número mínimo de cores necessárias?

DEFINIÇÃO 2 O *número cromático* de um grafo é o menor número de cores necessárias para a coloração deste grafo. O número cromático de um grafo G é indicado por $\chi(G)$. (Aqui, χ é a letra grega *chi*.)

Observe que perguntar o número cromático de um grafo planar é o mesmo que perguntar o número mínimo de cores necessárias para colorir um mapa planar de modo que duas regiões adjacentes não estejam associadas a uma mesma cor. Esta questão foi estudada por mais de 100 anos. A resposta é fornecida por um dos mais famosos teoremas da matemática.

TEOREMA 1 O TEOREMA DAS QUATRO CORES O número cromático de um grafo planar não é maior do que quatro.

Links

O Teorema das Quatro Cores foi originalmente proposto como uma conjectura na década de 1850. Ele foi finalmente demonstrado pelos matemáticos americanos Kenneth Appel e Wolfgang Haken em 1976. Antes de 1976, muitas demonstrações incorretas foram publicadas, em geral com erros difíceis de serem encontrados. Além disso, foram feitas muitas tentativas fúteis de construir contra-exemplos desenhando mapas que exigem mais do que quatro cores. (Demonstrar o Teorema das Cinco Cores não é difícil, veja o Exercício 36.)

Talvez a demonstração falaciosa mais conhecida em toda a matemática seja a demonstração incorreta do Teorema das Quatro Cores publicada em 1879 pelo advogado londrino e matemático amador Alfred Kempe. Os matemáticos aceitaram a demonstração dele como correta até 1890, quando Percy Heawood encontrou um erro que tornou o argumento de Kempe incompleto. Entretanto, a linha de raciocínio de Kempe acabou sendo a base da demonstração bem-sucedida feita por Appel e Haken. A demonstração deles se baseia em uma análise cuidadosa caso a caso feita pelo computador. Eles mostraram que se o Teorema das Quatro Cores fosse falso, teria que existir um contra-exemplo de aproximadamente 2 000 tipos diferentes, e então mostraram que nenhum desses tipos existia. Eles usaram mais de 1 000 horas de tempo de computação em sua demonstração. Essa demonstração gerou uma grande controvérsia, pois os computadores desempenharam um papel importante nela. Por exemplo, poderia haver um erro no programa de computador que levasse a resultados incorretos? O argumento deles era realmente uma demonstração, se dependia de uma saída não confiável de um programa de computador?

Observe que o Teorema das Quatro Cores se aplica apenas a grafos planares. Grafos não planares podem ter números cromáticos arbitrariamente grandes, como será mostrado no Exemplo 2.

Duas coisas são necessárias para mostrar que o número cromático de um grafo é k. Primeiro, precisamos mostrar que o grafo pode ser colorido com k cores. Isto pode ser feito construindo tal coloração. Segundo, devemos mostrar que o grafo não pode ser colorido usando menos que k cores. Os exemplos 1 a 4 ilustram como os números cromáticos podem ser encontrados.

EXEMPLO 1 Quais são os números cromáticos dos grafos G e H mostrados na Figura 3?

Exemplos Extras

Solução: O número cromático de G é, pelo menos, três, pois os vértices a , b e c devem estar associados a cores diferentes. Para ver se G pode ser colorido com três cores, associe vermelho a a, azul a b e verde a c. Então, d pode (e deve) ser colorido de vermelho, pois ele é adjacente a b e c. Além disso, e pode (e deve) ser colorido de verde, pois ele é adjacente apenas a vértices coloridos de vermelho e azul, e f pode (e deve) ser colorido de azul, pois é adjacente apenas a vértices coloridos de vermelho e verde. Finalmente, g pode (e deve) ser colorido de vermelho, pois

Links

ALFRED BRAY KEMPE (1849–1922) Kempe era um advogado e uma autoridade em leis eclesiásticas. Entretanto, tendo estudado matemática na Universidade de Cambridge, ele manteve o seu interesse por ela e, mais tarde, dedicou um tempo considerável à pesquisa matemática. Kempe deu contribuições à cinemática, o ramo da matemática que trata do movimento, e à lógica matemática. Entretanto, Kempe é mais lembrado pela sua demonstração falaciosa do Teorema das Quatro Cores.

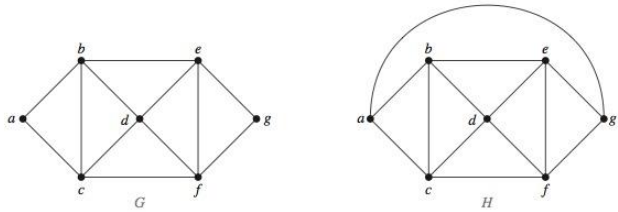


FIGURA 3 **Os Grafos Simples *G* e *H*.**

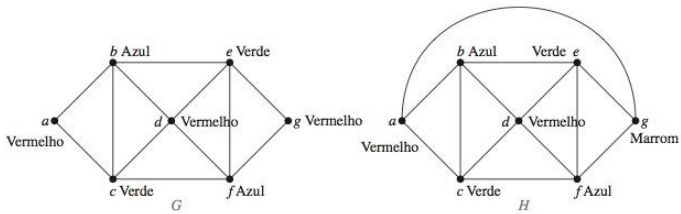


FIGURA 4 **Colorações dos Grafos *G* e *H*.**

ele é adjacente apenas a vértices coloridos de azul e verde. Isto produz uma coloração de G usando exatamente três cores. A Figura 4 exibe tal coloração.

O grafo H é obtido do grafo G adicionando uma aresta que conecta a e g. Qualquer tentativa de colorir H usando três cores deve seguir o mesmo raciocínio que o usado para colorir G, exceto no último estágio, quando todos os vértices diferentes de g tiverem sido coloridos. Então, como g é adjacente (em H) a vértices coloridos de vermelho, azul e verde, uma quarta cor, digamos marrom, precisa ser usada. Portanto, H tem um número cromático igual a 4. Uma coloração de H é mostrada na Figura 4. ◀

EXEMPLO 2 Qual é o número cromático de K_n?

Solução: Uma coloração de K_n pode ser construída usando n cores, associando uma cor diferente a cada vértice. Existe uma coloração que use menos cores? A resposta é não. Nenhum par de vértices pode estar associado à mesma cor, pois quaisquer dois vértices deste grafo são adjacentes. Logo, o número cromático de $K_n = n$. Ou seja, $\chi(K_n) = n$. (Lembre que K_n não é planar

---

OBSERVAÇÃO HISTÓRICA Em 1852, um ex-aluno de Augustus De Morgan, Francis Guthrie, observou que os condados na Inglaterra poderiam ser coloridos usando quatro cores, de modo que condados adjacentes nunca estivessem associados à mesma cor. Com base nesta evidência, ele conjecturou que o Teorema das Quatro Cores fosse verdade. Francis contou a seu irmão Frederick, naquela época um aluno de De Morgan, sobre este problema. Frederick, por sua vez, perguntou ao seu professor De Morgan sobre a conjectura do seu irmão. De Morgan ficou extremamente interessado neste problema e o popularizou entre a comunidade matemática. De fato, a primeira referência escrita à conjectura pode ser encontrada em uma carta de De Morgan a Sir William Rowan Hamilton. Embora De Morgan achasse que Hamilton fosse se interessar por este problema, Hamilton aparentemente não estava interessado nele, porque nada tinha a ver com os quaternions.

Links

OBSERVAÇÃO HISTÓRICA Apesar de uma demonstração mais simples do Teorema das Quatro Cores ter sido encontrada por Robertson, Sanders, Seymour e Thomas em 1996, reduzindo a parte computacional da demonstração ao exame de 633 configurações, nenhuma demonstração que não se baseie em amplo uso do computador ainda foi encontrada.

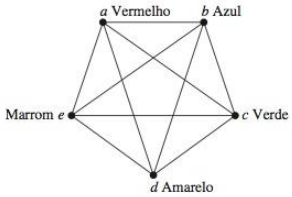


FIGURA 5 **Uma Coloração de K_5.**

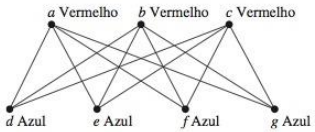


FIGURA 6 **Uma Coloração de $K_{3,4}$.**

quando $n \geq 5$, de modo que este resultado não contradiz o Teorema das Quatro Cores.) Uma coloração de K_5 usando cinco cores é mostrada na Figura 5. ◀

EXEMPLO 3 Qual é o número cromático do grafo bipartido completo $K_{m,n}$, em que m e n são inteiros positivos?

Solução: O número de cores necessárias pode parecer depender de m e n. Entretanto, como o Teorema 4 da Seção 9.2 nos diz, apenas duas cores são necessárias, pois $K_{m,n}$ é um grafo bipartido. Portanto, $\chi(K_{m,n}) = 2$. Isto significa que podemos colorir o conjunto dos m vértices com uma cor e o conjunto dos n vértices com uma segunda cor. Já que as arestas conectam apenas um vértice do conjunto dos m vértices a um vértice do conjunto dos n vértices, nenhum par de vértices adjacentes tem a mesma cor. Uma coloração de $K_{3,4}$ com duas cores é mostrada na Figura 6. ◀

EXEMPLO 4 Qual é o número cromático do grafo C_n, em que $n \geq 3$? (Lembre que C_n é o ciclo com n vértices.)

Solução: Consideraremos primeiro alguns casos individuais. Para começar, seja $n = 6$. Escolha um vértice e pinte-o de vermelho. Prossiga no sentido horário na descrição planar de C_6 mostrada na Figura 7. É necessário associar uma segunda cor, digamos azul, ao próximo vértice atingido. Continue no sentido horário; o terceiro vértice pode ser colorido de vermelho, o quarto vértice de azul e o quinto vértice de vermelho. Finalmente, o sexto vértice, que é adjacente ao primeiro, pode ser colorido de azul. Logo, o número cromático de C_6 é 2. A Figura 7 mostra a coloração construída aqui.

A seguir, seja $n = 5$ e considere C_5. Escolha um vértice e pinte-o de vermelho. Prosseguindo no sentido horário, é necessário associar uma segunda cor, digamos azul, ao próximo vértice atingido. Continuando no sentido horário, o terceiro vértice pode ser colorido de vermelho e o quarto vértice pode ser colorido de azul.

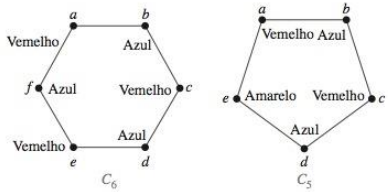


FIGURA 7 **Colorações de C_5 e C_6.**

O quinto vértice não pode ser colorido nem de vermelho nem de azul, pois ele é adjacente ao quarto vértice e ao primeiro vértice. Conseqüentemente, é necessária uma terceira cor para este vértice. Observe que também precisaríamos de três cores se tivéssemos colorido os vértices no sentido anti-horário. Assim, o número cromático de C_5 é 3. Uma coloração de C_5 usando três cores está mostrada na Figura 7.

Em geral, são necessárias duas cores para colorir C_n quando n é par. Para construir tal coloração, simplesmente escolha um vértice e pinte-o de vermelho. Prossiga em torno do grafo no sentido horário (usando uma representação planar do grafo), colorindo o segundo vértice de azul, o terceiro vértice de vermelho e assim por diante. O n-ésimo vértice pode ser colorido de azul, pois os dois vértices adjacentes a ele, a saber, o $(n-1)$-ésimo e o primeiro vértices, estão ambos coloridos de vermelho.

Quando n é ímpar e $n > 1$, o número cromático de C_n é 3. Para ver isso, escolha um vértice inicial. Para usar apenas duas cores, é necessário alternar as cores conforme o grafo é percorrido no sentido horário. Entretanto, o n-ésimo vértice atingido é adjacente a dois vértices de cores diferentes, a saber, o primeiro e o $(n-1)$-ésimo. Logo, deve ser usada uma terceira cor.

Mostramos que $\chi(C_n) = 2$ se n for um inteiro positivo par com $n \geq 4$ e $\chi(C_n) = 3$ se n for um inteiro positivo ímpar com $n \geq 3$. ◀

Links

Os melhores algoritmos conhecidos para determinar o número cromático de um grafo têm complexidade no tempo exponencial no pior caso (no número de vértices do grafo). Mesmo o problema de encontrar uma aproximação para o número cromático de um grafo é difícil. Foi mostrado que se houvesse um algoritmo com complexidade no tempo polinomial no pior caso que pudesse aproximar o número cromático de um grafo até um fator de 2 (isto é, construir um limitante que não fosse maior do que o dobro do número cromático do grafo), então também existiria um algoritmo com complexidade no tempo polinomial no pior caso para determinar o número cromático do grafo.

Aplicações da Coloração de Grafos

A coloração de grafos tem diversas aplicações a problemas que envolvem planejamento de cronogramas e distribuições. (Observe que já que não é conhecido nenhum algoritmo eficiente para a coloração de grafos, isto não leva a algoritmos eficientes para planejamento de cronograma e distribuições.) Serão dados aqui exemplos dessas aplicações. A primeira aplicação lida com planejamento de horário de exames finais.

EXEMPLO 5 **Horário dos Exames Finais** Como os exames finais em um universidade podem ser marcados de modo que nenhum estudante tenha dois exames ao mesmo tempo?

Solução: Este problema de planejamento de horário pode ser resolvido usando um modelo de grafo, com vértices representando cursos e com uma aresta entre dois vértices se existir um estudante comum nos cursos que eles representam. Cada posição na grade horária para um exame final é representada por uma cor diferente. Um horário dos exames corresponde a uma coloração do grafo associado.

Por exemplo, suponha que existam sete exames finais a serem marcados. Suponha que as disciplinas estejam numeradas de 1 a 7. Suponha que os seguintes pares de disciplinas tenham estudantes em comum: 1 e 2, 1 e 3, 1 e 4, 1 e 7, 2 e 3, 2 e 4, 2 e 5, 2 e 7, 3 e 4, 3 e 6, 3 e 7, 4 e 5, 4 e 6, 5 e 6, 5 e 7, e 6 e 7. Na Figura 8, é mostrado o grafo associado com este conjunto de disciplinas. Um horário consiste em uma coloração deste grafo.

Como o número cromático deste grafo é 4 (o leitor deve verificar isto), são necessárias quatro posições na grade horária. Uma coloração do grafo usando quatro cores e o horário associado estão mostrados na Figura 9. ◀

Considere agora uma aplicação na distribuição de canais de televisão.

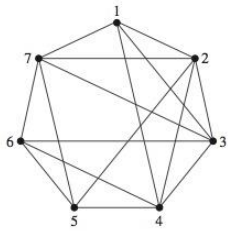
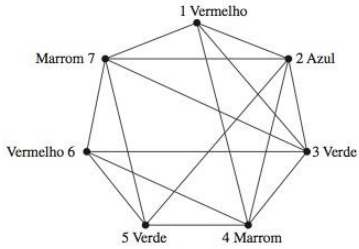


FIGURA 8 O Grafo Que Representa o Horário dos Exames Finais.



Período de tempo	Disciplina
I	1, 6
II	2
III	3, 5
IV	4, 7

FIGURA 9 Uso de uma Coloração para Marcar os Exames Finais.

EXEMPLO 6 **Designação de Freqüência** Os canais de televisão de 2 a 13 são designados a estações na América do Norte, de modo que nenhum par de estações a menos de 150 milhas pode operar no mesmo canal. Como a distribuição de canais pode ser modelada pela coloração de grafos?

Solução: Construa um grafo associando um vértice a cada estação. Dois vértices estão conectados por uma aresta se eles estiverem localizados a menos de 150 milhas um do outro. Uma distribuição de canais corresponde a uma coloração do grafo, em que cada cor representa um canal diferente. ◀

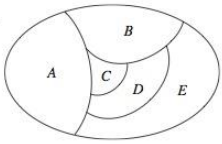
No Exemplo 7, é considerada uma aplicação da coloração de grafos a compiladores.

EXEMPLO 7 **Registros de Índices** Em compiladores eficientes, a execução de laços é acelerada quando variáveis usadas com freqüência são temporariamente armazenadas em registros de índice na unidade de processamento central, em vez de na memória regular. Para um dado laço, quantos registros de índice são necessários? Este problema pode ser tratado usando um modelo de coloração de grafos. Para montar o modelo, faça cada vértice do grafo representar uma variável no laço. Existe uma aresta entre dois vértices se as variáveis que eles representam precisarem ser armazenadas em registros de índice ao mesmo tempo durante a execução do laço. Assim, o número cromático do grafo fornece o número de registros de índice necessários, já que registros diferentes devem ser associados a variáveis quando os vértices que representam estas variáveis forem adjacentes no grafo. ◀

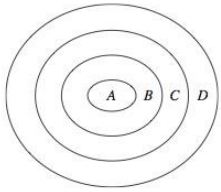
Exercícios

Nos exercícios 1 a 4, construa o grafo dual para o mapa mostrado. A seguir, encontre o número de cores necessárias para colorir o mapa de modo que nenhum par de regiões adjacentes tenha a mesma cor.

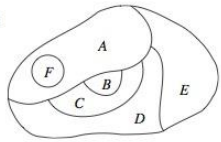
1.



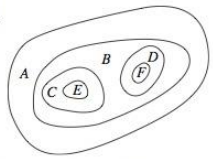
2.



3.



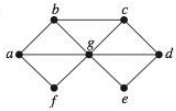
4.



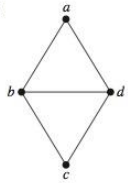
Nos exercícios 5 a 11, determine o número cromático do grafo dado.

5.

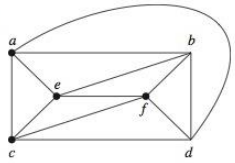
6.



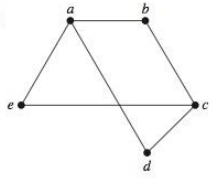
7.



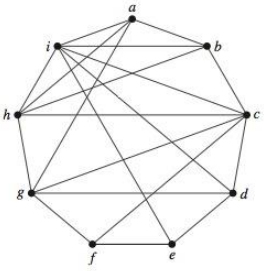
8.



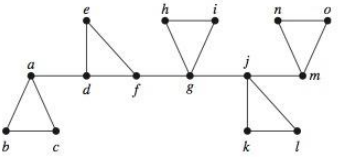
9.



10.



11.



12. Para os grafos nos exercícios 5 a 11, decida se é possível diminuir o número cromático pela remoção de um único vértice e de todas as arestas incidentes a ele.

13. Quais grafos têm número cromático 1?

14. Qual é o menor número de cores necessárias para colorir um mapa dos Estados Unidos? Não considere como adjacentes Estados que se encontram em uma quina. Suponha que Michigan seja uma região. Considere os vértices que representam o Alaska e o Havaí como vértices isolados.

15. Qual é o número cromático de W_n?

16. Mostre que um grafo simples que tem um ciclo com um número ímpar de vértices nele não pode ser colorido usando duas cores.

17. Faça o horário para os exames finais de Math 115, Math 116, Math 185, Math 195, CS 101, CS 102, CS 273 e CS 473, usando o menor número de posições diferentes na grade horária, se não existem estudantes matriculados simultaneamente em Math 115 e CS 473, Math 116 e CS 473, Math 195 e CS 101, Math 195 e CS 102, Math 115 e Math 116, Math 115 e Math 185 e Math 185 e Math 195, mas existem estudantes em todas as outras combinações de cursos.

18. Quantos canais diferentes são necessários para seis estações localizadas nas distâncias mostradas na tabela, se duas estações não puderem usar o mesmo canal quando estiverem a menos de 150 milhas uma da outra?

	1	*2*	*3*	*4*	*5*	*6*
1	—	85	175	200	50	100
2	85	—	125	175	100	160
3	175	125	—	100	200	250
4	200	175	100	—	210	220
4	50	100	200	210	—	100
6	100	160	250	220	100	—

19. O departamento de matemática tem seis comissões, cada uma se reunindo uma vez por mês. Quantos horários de reunião diferentes devem ser usados para garantir que nenhum membro tenha que participar de duas reuniões ao mesmo tempo, se os comitês são $C_1 = \{$Arlinghaus, Brand, Zaslavsky$\}$, $C_2 = \{$Brand, Lee, Rosen$\}$, $C_3 = \{$Arlinghaus, Rosen, Zaslavsky$\}$, $C_4 = \{$Lee, Rosen, Zaslavsky$\}$, $C_5 = \{$Arlinghaus, Brand$\}$ e $C_6 = \{$Brand, Rosen, Zaslavsky$\}$?

20. Um zoológico quer montar habitats naturais nos quais possa exibir seus animais. Infelizmente, alguns animais comerão alguns dos outros se tiverem oportunidade. Como um modelo de grafo e uma coloração podem ser usados para determinar o número de habitats diferentes necessários e a colocação dos animais nesses habitats?

Uma **coloração de arestas** de um grafo é uma associação de cores às arestas, de modo que arestas incidentes em um vértice comum estejam associadas a cores diferentes. O **número cromático de arestas** de um grafo é o menor número de cores que podem ser usadas em uma coloração de arestas de um grafo.

21. Encontre o número cromático de arestas em cada um dos grafos dos exercícios 5 a 11.

***22.** Encontre o número cromático de arestas de

a) K_n. **b)** $K_{m,n}$.

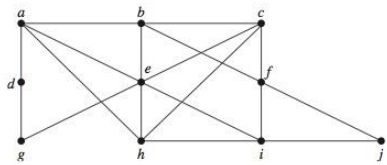
c) C_n. **d)** W_n.

23. Ocorrem sete variáveis em um laço de um programa de computador. As variáveis e os passos durante os quais elas devem ser armazenadas são *t*: passos 1 a 6; *u*: passo 2; *v*: passos 2 a 4; *w*: passos 1, 3 e 5; *x*: passos 1 e 6; *y*: passos 3 a 6; e *z*: passos 4 e 5. Quantos registros de índices diferentes são necessários para armazenar essas variáveis durante a execução?

24. O que pode ser dito sobre o número cromático de um grafo que tem K_n como um subgrafo?

Este algoritmo pode ser usado para colorir um grafo simples: Primeiro, liste os vértices $v_1, v_2, v_3, \cdots, v_n$ em ordem decrescente de grau, de modo que $\text{gr}(v_1) \geq \text{gr}(v_2) \geq \cdots \geq \text{gr}(v_n)$. Associe a cor 1 a v_1 e ao próximo vértice da lista que não seja adjacente a v_1 (se existir algum) e sucessivamente a cada vértice da lista que não seja adjacente a um vértice ao qual já tenha sido associada a cor 1. A seguir, associe a cor 2 ao primeiro vértice da lista que ainda não tenha sido colorido. Associe sucessivamente a cor 2 aos vértices da lista que ainda não tenham sido coloridos e que não sejam adjacentes aos vértices associados à cor 2. Se restarem vértices não coloridos, associe a cor 3 ao primeiro vértice da lista que ainda não esteja colorido e use a cor 3 para colorir sucessivamente os vértices que ainda não tenham sido coloridos e que não sejam adjacentes aos vértices associados à cor 3. Continue este processo até que todos os vértices tenham sido coloridos.

25. Construa uma coloração do grafo mostrado usando esse algoritmo.



***26.** Use pseudocódigo para descrever esse algoritmo de coloração.

***27.** Mostre que a coloração produzida nesse algoritmo pode usar mais cores do que é necessário para colorir um grafo.

Um grafo conexo *G* é dito **cromaticamente *k*-crítico** se o número cromático de *G* for *k*, mas para toda aresta *e* de *G*, o número cromático do grafo obtido pela remoção desta aresta de *G* é $k - 1$.

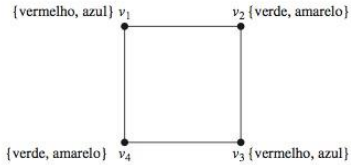
28. Mostre que C_n é cromaticamente 3-crítico sempre que *n* for um inteiro positivo ímpar, $n \geq 3$.

29. Mostre que W_n é cromaticamente 4-crítico sempre que *n* for um inteiro ímpar, $n \geq 3$.

30. Mostre que W_4 não é cromaticamente 3-crítico.

31. Mostre que se *G* for um grafo cromaticamente *k*-crítico, então o grau de todos os vértices de *G* é, pelo menos, $k - 1$.

Uma **coloração *k*-upla** de um grafo *G* é uma associação de um conjunto de *k* cores diferentes a cada um dos vértices de *G*, de modo que nenhum par de vértices adjacentes tenha associado a ele cores em comum. Indicamos por $\chi_k(G)$ o menor inteiro positivo *n* tal que *G* tenha um coloração *k*-upla usando *n* cores, por exemplo, $\chi_2(C_4) = 4$. Para ver isto, observe que usando apenas quatro cores podemos associar duas cores a cada vértice de C_4, como ilustrado, de modo que nenhum par de vértices adjacentes está associado a uma mesma cor. Além disso, menos do que quatro cores não são suficientes, pois os vértices v_1 e v_2 devem estar associados a duas cores cada um, e uma cor comum não pode ser associada a ambos, v_1 e v_2. (Para mais informação sobre coloração *k*-upla, veja [MiRo91].)



32. Encontre estes valores:

a) $\chi_2(K_3)$ **b)** $\chi_2(K_4)$ **c)** $\chi_2(W_4)$

d) $\chi_2(C_5)$ **e)** $\chi_2(K_{3,4})$ **f)** $\chi_3(K_5)$

***g)** $\chi_3(C_5)$ **h)** $\chi_3(K_{4,5})$

***33.** Sejam *G* e *H* os grafos mostrados na Figura 3. Determine

a) $\chi_2(G)$. **b)** $\chi_2(H)$.

c) $\chi_3(G)$. **d)** $\chi_3(H)$.

34. Qual é $\chi_k(G)$ se *G* for um grafo bipartido e *k* for um inteiro positivo?

35. Freqüências para telefones móveis (por rádio ou celular) são designadas por zonas. A cada zona é associado um conjunto de freqüências para serem usadas por veículos naquela zona. A mesma freqüência não pode ser usada em zonas nas quais a interferência seja um problema. Explique como uma coloração *k*-upla pode ser usada para associar *k* freqüências a cada zona de rádio móvel na região.

***36.** Mostre que todo gráfico planar *G* pode ser colorido usando seis ou menos cores. [*Dica:* Use indução matemática no número de vértices do grafo. Aplique o Corolário 2 da Sessão 9.7 para encontrar um vértice *v* com $\text{gr}(v) \leq 5$. Considere o subgrafo de *G* obtido excluindo *v* e todas as arestas incidentes nele].

****37.** Mostre que todo grafo planar G pode ser colorido usando cinco ou menos cores. [*Dica:* Use a sugestão fornecida no Exercício 36].

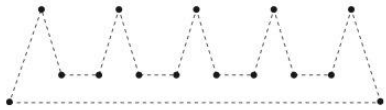
O famoso Problema da Galeria de Arte pergunta quantos guardas são necessários para ver todas as partes de uma galeria de arte, em que a galeria é o interior e a fronteira de um polígono com n lados. Para enunciar este problema mais precisamente, necessitamos de alguma terminologia. Um ponto x dentro ou na fronteira de um polígono simples P **cobre** ou **vê** um ponto y dentro de ou sobre P se todos os pontos no segmento de reta xy estiverem no interior ou na fronteira de P. Dizemos que um conjunto de pontos é um **conjunto de guarda** de um polígono simples P se para todo ponto y dentro de P ou na fronteira de P existir um ponto x neste conjunto de guarda que vê y. Indique por $G(P)$ o número mínimo de pontos necessários para guardar um polígono simples P. O **Problema da Galeria de Arte** pede a função $g(n)$, que é o valor máximo de $G(P)$ sobre todos os polígonos simples com n vértices. Isto é, $g(n)$ é o mínimo inteiro positivo para o qual é garantido que um polígono simples com n vértices pode ser guardado com $g(n)$ ou menos guardas.

38. Prove que $g(3) = 1$ e $g(4) = 1$ mostrando que todos os triângulos e quadriláteros podem ser guardados usando um ponto.

***39.** Mostre que $g(5) = 1$. Isto é, mostre que todos os pentágonos podem ser guardados usando um ponto. [*Dica:* Mostre que existem 0, 1 ou 2 vértices com um ângulo interior maior do que 180 graus e que, em cada caso, um guarda é suficiente.]

***40.** Mostre que $g(6) = 2$ primeiro usando os exercícios 38 e 39, bem como o Lema 1 da Sessão 4.2, para mostrar que $g(6) \leq 2$, e então encontre um hexágono simples para o qual são necessários dois guardas.

***41.** Mostre que $g(n) \geq \lfloor n/3 \rfloor$. [*Dica:* Considere o polígono com $3k$ vértices que parece um pente com k dentes, tal como o polígono com 15 lados mostrado aqui.]



***42.** Resolva o Problema da Galeria de Arte demonstrando o **Teorema da Galeria de Arte**, que afirma que são necessários no máximo $\lfloor n/3 \rfloor$ guardas para guardar o interior e a fronteira de um polígono simples com n vértices. [*Dica:* Use o Teorema 1 da Sessão 4.2 para triangular o polígono simples em $n - 2$ triângulos. Então, mostre que é possível colorir os vértices do polígono triangulado usando três cores, de modo que nenhum par de vértices adjacentes tenha a mesma cor. Use indução e o Exercício 23 da Sessão 4.2. Finalmente, ponha guardas em todos os vértices que estejam coloridos de vermelho, em que vermelho é a cor menos usada na coloração dos vértices. Mostre que colocar guardas nesses pontos é tudo que é necessário].

Termos-chave e Resultados

TERMOS

aresta não orientada: uma aresta associada a um conjunto $\{u, v\}$, em que u e v são vértices

aresta orientada: uma aresta associada a um par ordenado (u, v), em que u e v são vértices

arestas múltiplas: arestas distintas que conectam os mesmos vértices

arestas orientadas múltiplas: arestas orientadas distintas associadas ao mesmo par ordenado (u, v), em que u e v são vértices

laço: uma aresta que conecta um vértice a si mesmo

grafo não orientado: um conjunto de vértices e um conjunto de arestas não orientadas, cada um dos quais está associado a um conjunto de um ou dois desses vértices

grafo simples: um grafo não orientado sem arestas múltiplas ou laços

multigrafo: um grafo não orientado que pode conter arestas múltiplas, mas não laços

pseudografo: um grafo não orientado que pode conter arestas múltiplas e laços

grafo orientado: um conjunto de vértices junto com um conjunto de arestas orientadas, cada um dos quais está associado a um par ordenado de vértices

multigrafo orientado: um grafo com arestas orientadas que pode conter arestas orientadas múltiplas

grafo orientado simples: um grafo orientado sem laços ou arestas orientadas múltiplas

adjacente: dois vértices são adjacentes se existir uma aresta entre eles

incidente: uma aresta é incidente a um vértice se este vértice for uma extremidade daquela aresta

gr(v) (grau do vértice v em um grafo não orientado): o número de arestas incidentes em v com os laços contados duas vezes

gr$^-$(v) (grau de entrada do vértice v em um grafo com arestas orientadas): o número de arestas com v como seu vértice final

gr$^+$(v) (grau de saída do vértice v em um grafo com arestas orientadas): o número de arestas com v como seu vértice inicial

grafo não orientado subjacente a um grafo com arestas orientadas: o grafo não orientado obtido ignorando o sentido das arestas

K_n (grafo completo com n vértices): o grafo não orientado com n vértices, no qual cada par de vértices está ligado por uma aresta

grafo bipartido: um grafo com um conjunto de vértices que pode ser dividido em subconjuntos V_1 e V_2 tal que cada aresta conecta um vértice em V_1 a um vértice em V_2

$K_{m,n}$ (grafo bipartido completo): o grafo com o conjunto de vértices dividido em um subconjunto de m elementos e um subconjunto de n elementos tal que dois vértices estão conectados por uma aresta se e somente se uma estiver no primeiro subconjunto e a outra no segundo subconjunto

C_n (ciclo de tamanho n), $n \geq 3$: o grafo com n vértices $v_1, v_2, \cdots, v_n$ e arestas $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$

W_n (roda de tamanho n), $n \geq 3$: o grafo obtido a partir de C_n adicionando um vértice e arestas a partir deste vértice aos vértices originais de C_n

Q_n (n-cubo), $n \geq 1$: o grafo que tem 2^n seqüências de bits de comprimento n como seus vértices e arestas ligando todo par de seqüências de bits que diferem por exatamente um bit

vértice isolado: um vértice de grau zero

vértice pendente: um vértice de grau um

grafo regular: um grafo no qual todos os vértices têm o mesmo grau

subgrafo de um grafo $G = (V, E)$: um grafo (W, F), em que W é um subconjunto de V e F é um subconjunto de E

$G_1 \cup G_2$ (união de G_1 e G_2): o grafo $(V_1 \cup V_2, E_1 \cup E_2)$, em que $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$

matriz de adjacência: uma matriz que representa um grafo usando a adjacência dos vértices

matriz de incidência: uma matriz que representa um grafo usando a incidência de arestas e vértices

grafos simples isomorfos: os grafos simples $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ são isomorfos se existir uma correspondência biunívoca f de V_1 para V_2 tal que $\{f(v_1), f(v_2)\} \in E_2$ se e somente se $\{v_1, v_2\} \in E_1$ para todo v_1 e v_2 em V_1

invariante: uma propriedade que grafos isomorfos ou ambos têm ou ambos não têm.

caminho de u a v em um grafo não orientado: uma seqüência de arestas $e_1, e_2, \ldots, e_n$, em que e_i está associada a $\{x_i, x_{i+1}\}$ para $i = 0, 1, \ldots, n$, e em que $x_0 = u$ e $x_{n+1} = v$

caminho de u a v em um grafo com arestas orientadas: uma seqüência de arestas $e_1, e_2, \ldots, e_n$, em que e_i está associada a $\{x_i, x_{i+1}\}$ para $i = 0, 1, \ldots, n$, e em que $x_0 = u$ e $x_{n+1} = v$

caminho simples: um caminho que não contém uma aresta mais de uma vez

ciclo: um caminho de comprimento $n \geq 1$ que começa e termina no mesmo vértice

grafo conexo: um grafo não orientado com a propriedade que existe um caminho entre qualquer par de vértices

componente conexa de um grafo G: um subgrafo conexo maximal de G

grafo orientado fortemente conexo: um grafo orientado com a propriedade que existe um caminho orientado a partir de qualquer vértice até qualquer vértice

componente fortemente conexa de um grafo orientado G: um subgrafo fortemente conexo maximal de G

ciclo euleriano: um ciclo que contém toda aresta de um grafo exatamente uma vez

caminho euleriano: um caminho que contém todas as arestas de um grafo exatamente uma vez

caminho hamiltoniano: um caminho em um grafo simples que passa por todos os vértices exatamente uma vez

ciclo hamiltoniano: um ciclo em um grafo simples que passa por todos os vértices exatamente uma vez

grafo com pesos: um grafo com números associados às suas arestas

problema do caminho mais curto: o problema de determinar o caminho em um grafo com pesos tal que a soma dos pesos das arestas neste caminho é um mínimo de todos os caminhos entre os vértices especificados

problema do caixeiro-viajante: o problema que pede o ciclo de menor comprimento total que visite todos vértices do grafo exatamente uma vez

grafo planar: um grafo que pode ser desenhado no plano sem cruzamentos

regiões de uma representação de um grafo planar: as regiões nas quais o plano é dividido pela representação planar de um grafo

subdivisão elementar: a remoção de uma aresta $\{u, v\}$ de um grafo não orientado e a adição de um novo vértice w juntamente com as arestas $\{u, w\}$ e $\{w, v\}$

homeomorfos: dois grafos não orientados são homeomorfos se eles puderem ser obtidos de um mesmo grafo por uma seqüência de subdivisões elementares

coloração de grafos: uma associação de cores aos vértices de um grafo de modo que nenhum par de vértices adjacentes tenha a mesma cor

número cromático: o número mínimo de cores necessárias para colorir um grafo

RESULTADOS

Existe um ciclo euleriano em um multigrafo conexo se e somente se todo vértice tiver grau par.

Existe um caminho euleriano em um multigrafo conexo se e somente se no máximo dois vértices tiverem grau ímpar.

Algoritmo de Dijkstra: um procedimento para encontrar o caminho mais curto entre dois vértices em um grafo com pesos (veja a Seção 9.6).

Fórmula de Euler: $r = e - v + 2$, em que r, e e v são o número de regiões de uma representação planar, o número de arestas e o número de vértices, respectivamente, de um grafo planar.

Teorema de Kuratowski: Um grafo não é planar se e somente se contiver um subgrafo homeomorfo a $K_{3,3}$ ou K_5. (Demonstração além do escopo deste livro.)

O Teorema das Quatro Cores: Todo grafo planar pode ser colorido usando não mais que quatro cores. (Demonstração muito além do escopo deste livro!)

Questões de Revisão

1. **a)** Defina um grafo simples, um multigrafo, um pseudografo, um grafo orientado e um multigrafo orientado.
   b) Use um exemplo para mostrar como cada tipo de grafo na parte (a) pode ser usado em modelagem. Por exemplo, explique como modelar aspectos diferentes de uma rede de computadores ou rotas de avião.
2. Dê pelo menos quatro exemplos de como os grafos são usados em modelagem.

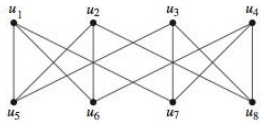
3. Qual é a relação entre a soma dos graus dos vértices em um grafo não orientado e o número de arestas neste grafo? Explique por que a relação é válida.
4. Por que deve existir um número par de vértices de grau ímpar em um grafo não orientado?
5. Qual é a relação entre a soma dos graus de entrada e a soma dos graus de saída dos vértices de um grafo orientado? Explique por que esta relação é válida.
6. Descreva as seguintes famílias de grafos.
   a) K_n, o grafo completo de n vértices
   b) $K_{m,n}$, o grafo bipartido completo de m e n vértices
   c) C_n, o ciclo com n vértices
   d) W_n, a roda de tamanho n
   e) Q_n, o n-cubo
7. Quantos vértices e quantas arestas existem em cada um dos grafos nas famílias da Questão 6?
8. **a)** O que é um grafo bipartido?
   b) Quais dos grafos K_n C_n e W_n são bipartidos?
   c) Como você pode determinar se um grafo não orientado é bipartido?
9. **a)** Descreva três métodos diferentes que podem ser usados para representar um grafo.
   b) Desenhe um grafo simples com, pelo menos, cinco vértices e oito arestas. Ilustre como ele pode ser representado usando os métodos que você descreveu na parte (a).
10. **a)** O que significa dois grafos simples serem isomorfos?
    b) O que queremos dizer com um invariante com relação a isomorfismos de grafos simples? Dê, pelo menos, cinco exemplos de tais invariantes.
    c) Dê um exemplo de dois grafos que tenham o mesmo número de vértices, arestas e graus dos vértices, mas que não sejam isomorfos.
    d) Existe um conjunto conhecido de invariantes que possa ser usado para determinar eficientemente se dois grafos simples são isomorfos?
11. **a)** O que significa um grafo ser conexo?
    b) O que são as componentes conexas de um grafo?
12. **a)** Explique como uma matriz de adjacência pode ser usada para representar um grafo.
    b) Como as matrizes de adjacência podem ser usadas para determinar se uma função do conjunto dos vértices de um grafo G para o conjunto de vértices de um grafo H é um isomorfismo?
    c) Como a matriz de adjacência de um grafo pode ser usada para determinar o número de caminhos de comprimento r, em que r é um inteiro positivo, entre dois vértices de um grafo?
13. **a)** Defina um ciclo euleriano e um caminho euleriano em um grafo não orientado.
    b) Descreva o famoso problema das pontes de Königsberg e explique como enunciá-lo em termos de um ciclo euleriano.
    c) Como pode ser determinado se um grafo não orientado tem um caminho euleriano?
    d) Como pode ser determinado se um grafo não orientado tem um ciclo euleriano?
14. **a)** Defina um ciclo hamiltoniano em um grafo simples.
    b) Dê algumas propriedades de um grafo simples que impliquem que ele não tem um ciclo hamiltoniano.
15. Dê exemplos de, pelo menos, dois problemas que possam ser resolvidos encontrando o caminho mais curto em um grafo com pesos.
16. **a)** Descreva o algoritmo de Dijkstra para encontrar o caminho mais curto entre dois vértices em um grafo com pesos.
    b) Desenhe um grafo com pesos com, pelo menos, 10 vértices e 20 arestas. Use o algoritmo de Dijkstra para encontrar o caminho mais curto entre dois vértices de sua escolha no grafo.
17. **a)** O que significa um grafo ser planar?
    b) Dê um exemplo de um grafo não planar.
18. **a)** Qual é a fórmula de Euler para um grafo planar?
    b) Como a fórmula de Euler para grafos planares pode ser usada para mostrar que um grafo simples não é planar?
19. Enuncie o Teorema de Kuratowski sobre grafos planares e explique como ele caracteriza quais grafos são planares.
20. **a)** Defina o número cromático de um grafo.
    b) Qual é o número cromático do grafo K_n quando n é um inteiro positivo?
    c) Qual é o número cromático do grafo C_n quando n é um inteiro positivo maior do que 2?
    d) Qual é o número cromático do grafo $K_{m,n}$ quando m e n são inteiros positivos?
21. Enuncie o Teorema das Quatro Cores. Existem grafos que não podem ser coloridos com quatro cores?
22. Explique como a coloração de grafos pode ser usada em modelagem. Use, pelo menos, dois exemplos diferentes.

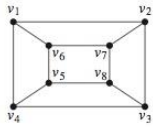
Exercícios Complementares

1. Quantas arestas tem um grafo 50-regular com 100 vértices?
2. Quantos subgrafos não isomorfos tem K_3?

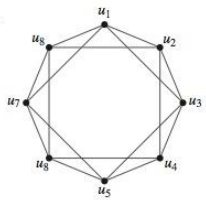
Nos exercícios 3 a 5, determine se o par de grafos dado é isomorfo.

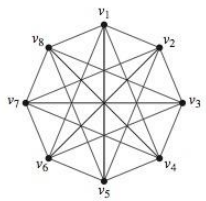
3.



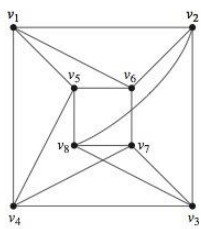


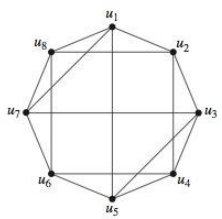
4.





***5.**





O **grafo *m*-partido completo** $K_{n_1,n_2,\ldots,n_m}$ tem vértices divididos em m subconjuntos de $n_1, n_2, \ldots, n_m$ elementos cada, e os vértices são adjacentes se e somente se estiverem em subconjuntos diferentes da partição.

6. Desenhe estes grafos.

a) $K_{1,2,3}$ **b)** $K_{2,2,2}$ **c)** $K_{1,2,2,3}$

***7.** Quantos vértices e quantas arestas tem o grafo m-partido completo $K_{n_1,n_2,\ldots,n_m}$?

***8.** **a)** Demonstre ou mostre que não é válido que sempre existem dois vértices com o mesmo grau em um grafo simples finito que tenha pelo menos dois vértices.

b) Faça o mesmo que na parte (a) para multigrafos finitos.

Seja $G = (V, E)$ um grafo simples. O **subgrafo induzido** por um subconjunto W do conjunto de vértices V é o grafo (W, F), no qual o conjunto de arestas F contém uma aresta em E se e somente se ambas as extremidades desta aresta estiverem em W.

9. Considere o grafo mostrado na Figura 3 da Seção 9.4. Encontre o subgrafo induzido por

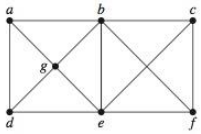
a) $\{a, b, c\}$. **b)** $\{a, e, g\}$.

c) $\{b, c, f, g, h\}$.

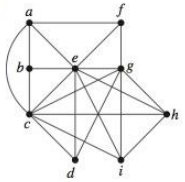
10. Seja n um inteiro positivo. Mostre que um subgrafo induzido por um conjunto não vazio do conjunto de vértices de K_n é um grafo completo.

Um **clique** em um grafo não orientado simples é um subgrafo completo que não está contido em nenhum subgrafo completo maior. Nos exercícios 11 a 13, encontre todos os cliques do grafo mostrado.

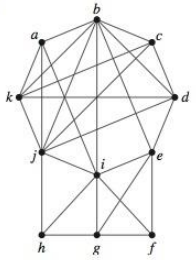
11.



12.

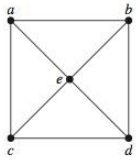


13.

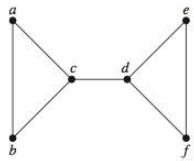


Um **conjunto dominante** de vértices em um grafo simples é um conjunto de vértices tal que todos os outros vértices são adjacentes a, pelo menos, um vértice deste conjunto. Um conjunto dominante com o menor número de vértices é chamado de **conjunto dominante mínimo**. Nos exercícios 14 a 16, encontre um conjunto dominante mínimo para o grafo dado.

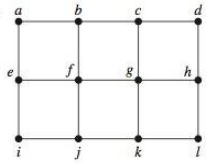
14.



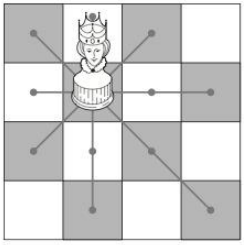
15.



16.



Um grafo simples pode ser usado para determinar o número mínimo de rainhas em um tabuleiro de xadrez que controlem o tabuleiro todo. Um tabuleiro de xadrez $n \times n$ tem n^2 quadrados em uma configuração $n \times n$. Uma rainha em uma dada posição controla todos os quadrados na mesma linha, na mesma coluna e nas duas diagonais que contêm este quadrado, como ilustrado abaixo. O grafo simples apropriado tem n^2 vértices, um para cada quadrado, e dois vértices são adjacentes se uma rainha no quadrado representado por um dos vértices controlar o quadrado representado pelo outro vértice.



Os Quadrados Controlados por uma Rainha

17. Construa o grafo simples que representa o tabuleiro $n \times n$ com arestas que representam o controle de quadrados por rainhas para

a) $n = 3$. **b)** $n = 4$.

18. Explique como o conceito de conjunto dominante mínimo se aplica ao problema de determinar o número mínimo de rainhas que controlam um tabuleiro $n \times n$.

****19.** Encontre o número mínimo de rainhas que controlam um tabuleiro $n \times n$ para

a) $n = 3$. **b)** $n = 4$. **c)** $n = 5$.

20. Suponha que G_1 e H_1 sejam isomorfos e que G_2 e H_2 sejam isomorfos. Demonstre ou mostre que não é válido que $G_1 \cup G_2$ e $H_1 \cup H_2$ são isomorfos.

21. Mostre que cada uma destas propriedades é um invariante que ou ambos grafos simples isomorfos têm ou ambos não têm.

a) conexidade

b) a existência de um ciclo hamiltoniano

c) a existência de um ciclo euleriano

d) ter número de cruzamentos C

e) ter n vértices isolados

f) ser bipartido

22. Como pode ser encontrada a matriz de adjacência de $\overline{G}$ a partir da matriz de adjacência de G, se G é um grafo simples?

23. Quantos grafos simples, bipartidos, conexos, não isomorfos existem com quatro vértices?

***24.** Quantos grafos conexos, simples, não isomorfos, com cinco vértices existem

a) sem nenhum vértice de grau maior do que dois?

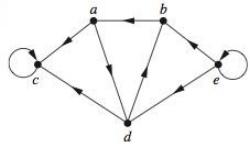
b) com número cromático igual a quatro?

c) que não são planares?

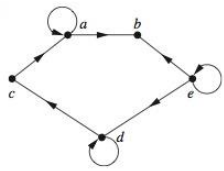
Um grafo orientado é **auto-reverso** se ele for isomorfo a seu reverso.

25. Determine se os seguintes grafos são auto-reversos.

a)



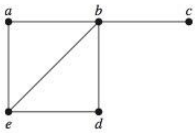
b)



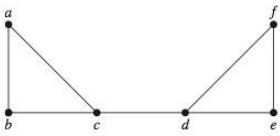
26. Mostre que se o grafo orientado G for auto-reverso e se H for um grafo orientado isomorfo a G, então H também é auto-reverso.

Uma **orientação** de um grafo simples não orientado é uma associação de sentidos a suas arestas de modo que o grafo orientado resultante seja fortemente conexo. Quando existir uma orientação de um grafo não orientado, este grafo é dito **orientável**. Nos exercícios 27 a 29, determine se o grafo simples dado é orientável.

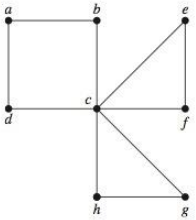
27.



28.



29.



30. Como o tráfego está se tornando pesado na parte central de uma cidade, os engenheiros de tráfego estão planejando mudar todas as ruas, que atualmente são de mão dupla, para ruas de mão única. Explique como modelar este problema.

***31.** Mostre que um grafo não é orientável se ele tiver uma aresta de corte.

Um **torneio** é um grafo orientado simples tal que se u e v forem vértices distintos do grafo, exatamente um entre (u, v) e (v, u) é uma aresta do grafo.

32. Quantos torneios diferentes existem com n vértices?

33. Qual é a soma dos graus de entrada e de saída de um vértice em um torneio?

***34.** Mostre que todo torneio tem um caminho hamiltoniano.

35. Dadas duas galinhas em um bando, uma delas é dominante. Isto define a **ordem pecking** (ato de bicar) no bando. Como um torneio pode ser usado para modelar a ordem pecking?

36. Suponha que G seja um multigrafo conexo com $2k$ vértices de grau ímpar. Mostre que existem k subgrafos que têm G como sua união, em que cada um destes subgrafos tem um caminho euleriano e nenhum par destes subgrafos tem uma aresta em comum. [*Dica:* Adicione k arestas ao grafo conectando pares de vértices de grau ímpar e use um ciclo euleriano neste grafo maior.]

***37.** Seja G um grafo simples com n vértices. A **largura de banda** de G, indicada por $B(G)$, é o mínimo, sobre todas as permutações, $a_1, a_2, \ldots, a_n$ dos vértices de G, de $\max\{|i-j|\}$ a_i e a_j são adjacentes}. Isto é, a largura de banda é o mínimo sobre todas as listagens de vértices da diferença máxima nos índices associados a vértices adjacentes. Encontre as larguras de bandas destes grafos.

a) K_5 **b)** $K_{1,3}$ **c)** $K_{2,3}$

d) $K_{3,3}$ **e)** Q_3 **f)** C_5

***38.** A **distância** entre dois vértices distintos v_1 e v_2 de um grafo simples conexo é o comprimento (número de arestas) do caminho mais curto entre v_1 e v_2. O **raio** de um grafo é o mínimo entre todos os vértices v da distância máxima de v a um outro vértice. O **diâmetro** de um grafo é a distância máxima entre dois vértices distintos. Encontre o raio e o diâmetro de

a) K_6. **b)** $K_{4,5}$. **c)** Q_3. **d)** C_6.

***39.** **a)** Mostre que se o diâmetro de um grafo simples G for, pelo menos, quatro, então o diâmetro de seu complementar $\overline{G}$ não é mais do que dois.

b) Mostre que se o diâmetro de um grafo simples G for, pelo menos, três, então o diâmetro de seu complementar $\overline{G}$ não é mais do que três.

***40.** Suponha que um multigrafo tenha $2m$ vértices de grau ímpar. Mostre que qualquer ciclo que contenha todas as arestas do grafo deve conter, pelo menos, m arestas mais de uma vez.

41. Encontre o segundo caminho mais curto entre os vértices a e z na Figura 3 da Seção 9.6.

42. Descubra um algoritmo para encontrar o segundo caminho mais curto entre dois vértices em um grafo com pesos conexo simples.

43. Encontre o menor caminho entre os vértices a e z que passa pelo vértice e no grafo com pesos na Figura 4 da Seção 9.6.

44. Descubra um algoritmo para encontrar o caminho mais curto entre dois vértices em um grafo com pesos conexo simples que passe por um terceiro vértice especificado.

***45.** Mostre que se G for um grafo simples com, pelo menos, 11 vértices, então ou G ou $\overline{G}$, o complementar de G, não é planar.

Um conjunto de vértices em um grafo é dito **independente** se nenhum par de vértices neste conjunto for adjacente. O **número de independência** de um grafo é o número máximo de vértices em um conjunto independente de vértices neste grafo.

***46.** Qual é o número de independência de

a) K_n? **b)** C_n? **c)** Q_n? **d)** $K_{m,n}$?

47. Mostre que o número de vértices em um grafo simples é menor que ou igual ao produto do número de independência e do número cromático do grafo.

48. Mostre que o número cromático de um grafo é menor que ou igual a $v - i + 1$, em que v é o número de vértices no grafo e i é o número de independência do grafo.

49. Suponha que, para gerar um grafo simples aleatório com n vértices, primeiro escolhemos um número real p com $0 \leq p \leq 1$. Para cada um dos $C(n, 2)$ pares de vértices distintos, geramos um número aleatório x entre 0 e 1. Se $0 \leq x \leq p$, conectamos estes dois vértices por uma aresta; caso contrário, estes dois vértices não são conectados.

a) Qual é a probabilidade de que seja gerado um grafo com m arestas, em que $0 \leq m \leq C(n, 2)$?
b) Qual é o número esperado de arestas em um grafo gerado aleatoriamente com n vértices se cada aresta for incluída com probabilidade p?
c) Mostre que se $p = 1/2$, então é igualmente provável gerar todo grafo simples com n vértices.

Uma propriedade mantida sempre que arestas adicionais forem acrescentadas a um grafo simples (sem acrescentar vértices) é dita **monótona crescente**, e uma propriedade que é mantida sempre que arestas forem removidas de um grafo simples (sem remover vértices) é dita **monótona decrescente**.

50. Para cada uma destas propriedades, determine se ela é monótona crescente e determine se é monótona decrescente.
a) O grafo G é conexo.
b) O grafo G não é conexo.
c) O grafo G tem um ciclo euleriano.
d) O grafo G tem um ciclo hamiltoniano.
e) O grafo G é planar.
f) O grafo G tem número cromático quatro.
g) O grafo G tem raio três.
h) O grafo G tem diâmetro três.

51. Mostre que a propriedade de grafos P é monótona crescente se e somente se a propriedade de grafos Q é monótona decrescente, em que a propriedade Q é não ter a propriedade P.

****52.** Suponha que P seja uma propriedade monótona crescente de grafos simples. Mostre que a probabilidade de um grafo aleatório com n vértices ter a propriedade P é uma função monótona não decrescente de p, a probabilidade de uma aresta ser escolhida para estar no grafo.

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dados os pares de vértices associados às arestas de um grafo não orientado, encontre o grau de cada vértice.
2. Dados os pares ordenados de vértices associados às arestas de um grafo orientado, determine o grau de entrada e o grau de saída de cada vértice.
3. Dada a lista das arestas de um grafo simples, determine se o grafo é bipartido.
4. Dados os pares de vértices associados às arestas de um grafo, construa uma matriz de adjacência do grafo. (Produza uma versão que funcione quando laços, arestas múltiplas ou arestas orientadas estiverem presentes.)
5. Dada uma matriz de adjacência de um grafo, liste as arestas deste grafo e dê o número de vezes que cada aresta aparece.
6. Dados os pares de vértices associados às arestas de um grafo não orientado e o número de vezes que cada aresta aparece, construa uma matriz de incidência deste grafo.
7. Dada uma matriz de incidência de um grafo não orientado, liste suas arestas e dê o número de vezes que cada aresta aparece.
8. Dado um inteiro positivo n, gere um grafo não orientado produzindo uma matriz de adjacência para o grafo, de modo que todos os grafos simples sejam igualmente prováveis de ser gerados.
9. Dado um inteiro n positivo, gere um grafo orientado produzindo uma matriz de adjacência para o grafo, de modo que todos os grafos orientados sejam igualmente prováveis de ser gerados.
10. Dada as listas das arestas de dois grafos simples com, no máximo, seis vértices, determine se os grafos são isomorfos.
11. Dada a matriz de adjacência de um grafo e um inteiro positivo n, encontre o número de caminhos de comprimento n entre dois vértices. (Produza uma versão que funcione para grafos orientados e não orientados.)

***12.** Dada a lista das arestas de um grafo simples, determine se ele é conexo e encontre o número de componentes conexas se ele não for conexo.

13. Dados os pares de vértices associados às arestas de um multigrafo, determine se ele tem um ciclo euleriano e, em caso negativo, se tem um caminho euleriano. Construa um caminho ou ciclo euleriano, caso exista.

***14.** Dados os pares ordenados de vértices associados às arestas de um multigrafo orientado, construa um caminho euleriano ou um ciclo euleriano, se tal caminho ou ciclo existir.

****15.** Dada a lista das arestas de um grafo simples, produza um ciclo hamiltoniano ou determine que o grafo não tem tal ciclo.

****16.** Dada a lista das arestas de um grafo simples, produza um caminho hamiltoniano ou determine que o grafo não tem tal caminho.

17. Dada a lista das arestas e seus pesos de um grafo simples, conexo, com pesos e dois vértices deste grafo, encontre o comprimento do caminho mais curto entre eles usando o algoritmo de Dijkstra. Além disso, determine um caminho mais curto.
18. Dada a lista das arestas de um grafo não orientado, encontre uma coloração deste grafo usando o algoritmo dado nos exercícios da Seção 9.8.
19. Dada uma lista dos estudantes e das disciplinas nas quais estão matriculados, construa um horário para os exames finais.
20. Dadas as distâncias entre pares de estações de televisão, associe freqüências a estas estações.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Exiba todos os grafos simples com quatro vértices.
2. Exiba um conjunto completo de grafos simples, não isomorfos com seis vértices.
3. Exiba um conjunto completo de grafos orientados, não isomorfos com quatro vértices.
4. Gere aleatoriamente 10 grafos simples diferentes, cada um com até 20 vértices, de modo que tais grafos sejam igualmente prováveis de serem gerados.
5. Construa um código de Gray no qual as palavras do código sejam cadeias de bit de comprimento seis.
6. Construa um passeio de cavalo em tabuleiros de xadrez de vários tamanhos.
7. Determine se cada um dos grafos que você gerou no Exercício 4 deste conjunto de exercícios é planar. Se você puder, determine a espessura de cada um dos grafos que não são planares.
8. Determine se cada um dos grafos que você gerou no Exercício 4 deste conjunto de exercícios é conexo. Se um grafo não for conexo, determine o número de componentes conexas do grafo.
9. Gere aleatoriamente um grafo simples com 10 vértices. Pare quando você tiver construído um que tenha um ciclo euleriano. Exiba um ciclo euleriano nesse grafo.
10. Gere aleatoriamente grafos simples com 10 vértices. Pare quando você tiver construído um que tenha um ciclo hamiltoniano. Exiba um ciclo hamiltoniano nesse grafo.
11. Encontre o número cromático de cada um dos grafos que você gerou no Exercício 4 deste conjunto de exercícios.
**12. Encontre o caminho mais curto que um caixeiro-viajante pode tomar para visitar cada uma das capitais dos 50 estados dos Estados Unidos, viajando por ar entre as cidades, em uma linha reta.
13. Estime a probabilidade de que um grafo simples, gerado aleatoriamente com n vértices, seja conexo para cada inteiro positivo n que não exceda 10, gerando um conjunto de grafos simples aleatórios e determinando se cada um deles é conexo.
**14. Trabalhe no problema de determinar se o número de cruzamentos de $K(7, 7)$ é 77, 79 ou 81. É sabido que ele é igual a um destes três valores.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Descreva as origens e o desenvolvimento da teoria dos grafos anterior ao ano de 1900.
2. Discuta as aplicações da teoria dos grafos ao estudo de ecossistemas.
3. Discuta as aplicações da teoria dos grafos à sociologia e à psicologia.
4. Discuta o que podemos aprender investigando as propriedades do grafo da Web.
5. Descreva algoritmos para desenhar um grafo em papel ou em uma janela, dados os vértices e as arestas do grafo. Que considerações aparecem ao se desenhar um grafo, de modo que ele tenha a melhor aparência para entender suas propriedades?
6. Quais são alguns dos recursos que um software deveria ter para entrar, exibir e manipular grafos? Quais desses recursos os softwares disponíveis têm?
7. Descreva alguns dos algoritmos disponíveis para determinar se dois grafos são isomorfos e a complexidade computacional destes algoritmos. Qual é o mais eficiente destes algoritmos conhecidos atualmente?
8. Descreva como caminhos eulerianos podem ser usados para ajudar a determinar seqüências de DNA.
9. Defina as *seqüências de Bruijn* e discuta como elas aparecem nessas aplicações. Explique como as seqüências de Bruijn podem ser construídas usando ciclos eulerianos.
10. Descreva o *problema do carteiro chinês* e explique como resolvê-lo.
11. Descreva algumas das condições diferentes que implicam que um grafo tem um ciclo hamiltoniano.
12. Descreva algumas das estratégias e algoritmos usados para resolver o problema do caixeiro-viajante.
13. Descreva diversos algoritmos diferentes para determinar se um grafo é planar. Qual é a complexidade computacional de cada um destes algoritmos?
14. Em modelagem, grafos com escalas muito grandes de integração (VLSI) algumas vezes estão imersos em um livro, com os vértices no dorso do livro e arestas nas páginas. Defina o *número de livro* de um grafo e determine os números de livros de diversos grafos, incluindo K_n para $n = 3, 4, 5$ e 6.
15. Discuta a história do Teorema das Quatro Cores.
16. Descreva o papel que os computadores desempenharam na demonstração do Teorema das Quatro Cores. Como podemos ter certeza de que uma demonstração baseada em um computador está correta?
17. Descreva e compare diversos algoritmos diferentes para colorir um grafo em termos de saber se eles produzem uma coloração com o menor número possível de cores e em termos de sua complexidade.
18. Explique como grafos multicoloridos podem ser usados em diversos modelos diferentes.
19. Explique como a teoria dos grafos aleatórios pode ser usada em demonstrações não construtivas da existência de grafos com certas propriedades.

CAPÍTULO

10 Árvores

10.1 Introdução às Árvores
10.2 Aplicações de Árvores
10.3 Percurso em Árvores
10.4 Árvores Geradoras
10.5 Árvores Geradoras Mínimas

Um grafo conexo que não contenha nenhum ciclo simples é chamado de árvore. As árvores são usadas desde 1857, quando o matemático inglês Arthur Cayley as usava para contar certos tipos de compostos químicos. Desde aquela época, as árvores têm sido usadas para resolver problemas em uma ampla variedade de disciplinas, como os exemplos deste capítulo mostrarão.

As árvores são particularmente úteis na ciência da computação, na qual são usadas em um amplo espectro de algoritmos. Por exemplo, são usadas para construir algoritmos eficientes para localizar itens em uma lista. Elas podem ser usadas em algoritmos como a codificação de Huffman, que constrói códigos eficientes e economiza custo na transmissão e no armazenamento de dados. As árvores podem ser usadas para estudar jogos como xadrez e damas e podem ajudar a determinar estratégias ganhadoras para esses jogos. As árvores podem ser usadas para modelar procedimentos implementados com uma seqüência de decisões. Construir estes modelos pode ajudar a determinar a complexidade computacional dos algoritmos que se baseiam em uma seqüência de decisões, tais como os algoritmos de ordenação.

Procedimentos para a construção de árvores que contenham todos os vértices de um grafo, incluindo busca em profundidade e busca em largura, podem ser usados para explorar sistematicamente os vértices de um grafo. A exploração dos vértices de um grafo por busca em profundidade, também conhecido por *backtracking*, permite a busca sistemática de soluções de uma ampla variedade de problemas, tal como a determinação de como oito rainhas podem ser colocadas em um tabuleiro de xadrez de modo que nenhuma rainha possa atacar qualquer outra.

Podemos associar pesos às arestas de uma árvore para modelar diversos problemas. Por exemplo, usando árvores com peso podemos desenvolver algoritmos para construir redes que contenham o conjunto mais barato de linhas telefônicas que ligam diferentes nós da rede.

10.1 Introdução às Árvores

Links

No Capítulo 9 mostramos como os grafos podem ser utilizados para modelar e resolver muitos problemas. Neste capítulo, nos concentraremos em um tipo particular de grafo chamado **árvore**, que é assim chamado porque esses grafos se parecem com árvores. Por exemplo, *árvores de família* são grafos que representam mapas genealógicos. Árvores de família usam vértices para representar os membros de uma família e arestas para representar relações de progenitura. A árvore de família dos membros masculinos da família Bernoulli de matemáticos suíços é mostrada na Figura 1. O grafo não orientado que representa uma árvore de família (restrito a pessoas de apenas um sexo e sem procriação consangüínea) é um exemplo de uma árvore.

DEFINIÇÃO 1 Uma *árvore* é um grafo não orientado conexo sem nenhum ciclo simples.

Como uma árvore não pode ter um ciclo simples, não pode conter arestas múltiplas ou laços. Portanto, qualquer árvore deve ser um grafo simples.

EXEMPLO 1 Quais dos grafos mostrados na Figura 2 são árvores?

Solução: G_1 e G_2 são árvores, pois ambos são grafos conexos sem ciclos simples. G_3 não é uma árvore, pois e, b, a, d e e é um ciclo simples neste grafo. Finalmente, G_4 não é uma árvore, pois não é conexo. ◄

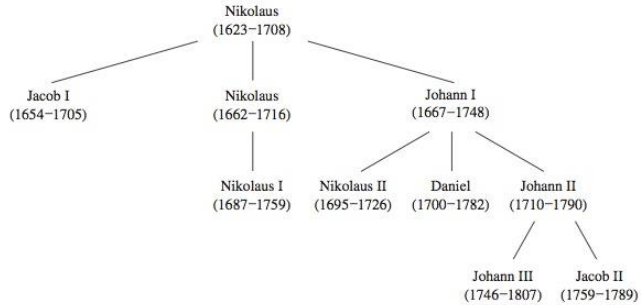


FIGURA 1 A Família Bernoulli de Matemáticos.

Qualquer grafo conexo que não contenha nenhum ciclo simples é uma árvore. E os grafos que não contêm nenhum ciclo simples, mas que não são necessariamente conexos? Esses grafos são chamados de **florestas** e têm a propriedade que cada uma de suas componentes conexas é uma árvore. A Figura 3 mostra uma floresta.

As árvores são freqüentemente definidas como grafos não orientados com a propriedade de que existe um único caminho simples entre todo par de vértices. O Teorema 1 mostra que esta definição alternativa é equivalente à nossa definição.

TEOREMA 1 Um grafo não orientado é uma árvore se e somente se existir um único caminho simples entre quaisquer dois de seus vértices.

Demonstração: Suponha primeiro que T seja uma árvore. Então, T é um grafo conexo sem ciclos simples. Sejam x e y dois vértices de T. Como T é conexo, pelo Teorema 1 da Seção 9.4 existe um caminho simples entre x e y. Além disso, esse caminho deve ser único, pois se existisse um segundo desses caminhos, o caminho formado pela combinação do primeiro caminho de x a y seguido pelo caminho de y a x obtido pela inversão da ordem do segundo caminho de x a y formaria um

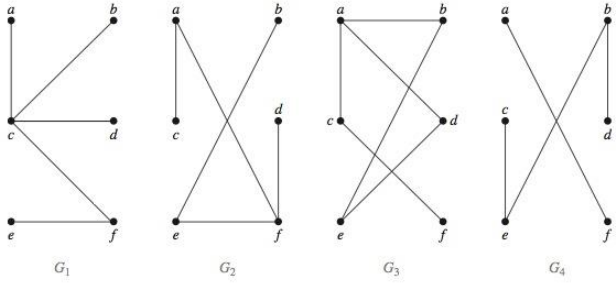


FIGURA 2 Exemplos de Árvores e de Grafos que Não São Árvores.

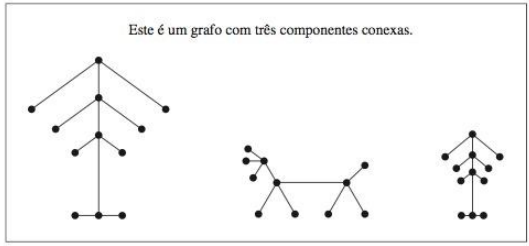


FIGURA 3 **Exemplo de uma Floresta.**

ciclo. Isso implica, usando o Exercício 49 da Seção 9.4, que existe um ciclo simples em T. Portanto, existe um único caminho simples entre quaisquer dois vértices de uma árvore.

Suponha agora que exista um único caminho simples entre quaisquer dois vértices de um grafo T. Então T é conexo, pois existe um caminho entre quaisquer dois de seus vértices. Além disso, T não pode ter ciclos simples. Para ver isso, suponha que T tenha um ciclo simples que contenha os vértices x e y. Então, existiriam dois caminhos simples entre x e y, pois o ciclo simples é formado de um caminho simples de x a y e um segundo caminho simples de y a x. Portanto, um grafo com um único caminho simples entre quaisquer dois vértices é uma árvore. ◁

Em muitas aplicações de árvores, um vértice particular da árvore é escolhido como **raiz**. Uma vez que especifiquemos uma raiz, podemos associar um sentido a cada aresta como explicamos a seguir. Como existe um único caminho da raiz a cada vértice do grafo (pelo Teorema 1), orientamos cada aresta no sentido oposto da raiz. Assim, uma árvore com sua raiz produz um grafo orientado chamado de **árvore com raiz** (ou **árvore enraizada**).

DEFINIÇÃO 2 Uma *árvore com raiz* (ou *árvore enraizada*) é uma árvore na qual um vértice tenha sido escolhido como raiz e toda aresta esteja orientada no sentido que se afasta da raiz.

As árvores com raiz também podem ser definidas recursivamente. Consulte a Seção 4.3 para ver como isso pode ser feito. Podemos transformar uma árvore sem raiz em uma árvore com raiz, escolhendo qualquer vértice como raiz. Observe que escolhas diferentes de raízes produzem árvores com raízes diferentes. Por exemplo, a Figura 4 mostra árvores enraizadas formadas pela escolha de a como a raiz e de c como a raiz, respectivamente, na árvore T. Usualmente desenhamos uma árvo-

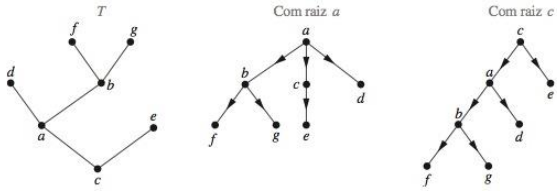


FIGURA 4 **Uma Árvore e Árvores Enraizadas Formadas pela Escolha de Duas Raízes.**

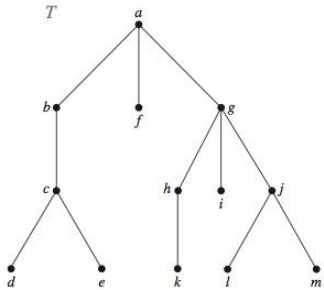


FIGURA 5 Uma Árvore Enraizada *T*.

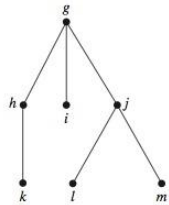


FIGURA 6 A Subárvore Enraizada em *g*.

re enraizada com sua raiz no topo do grafo. As flechas que indicam os sentidos das arestas em uma árvore enraizada podem ser omitidas, pois a escolha da raiz determina os sentidos das arestas.

A terminologia para árvores tem origem botânica e genealógica. Suponha que T seja uma árvore com raiz. Se v for um vértice em T diferente da raiz, o **pai** de v é o único vértice u tal que existe uma aresta orientada de u a v (o leitor deve mostrar que esse vértice é único). Quando u for o pai de v, v é chamado de **filho** de u. Vértices com o mesmo pai são chamados de **irmãos**. Os **ancestrais** de um vértice diferente da raiz são os vértices no caminho da raiz até este vértice, excluindo o próprio vértice e incluindo a raiz (isto é, seu pai, o pai de seu pai e assim por diante, até que a raiz seja atingida). Os **descendentes** de um vértice v são os vértices que têm v como um ancestral. Um vértice de uma árvore é chamado de **folha** se não tiver filhos. Os vértices que têm filhos são chamados de **vértices internos**. A raiz é um vértice interno a menos que ela seja o único vértice do grafo, em cujo caso é também uma folha.

Se a for um vértice em uma árvore, a **subárvore** com a como sua raiz é o subgrafo da árvore que consiste em a e seus descendentes e em todas as arestas incidentes sobre estes descendentes.

EXEMPLO 2 Na árvore enraizada T (com raiz a) mostrada na Figura 5, encontre o pai de c, os filhos de g, os irmãos de h, todos os ancestrais de e, todos os descendentes de b, todos os vértices internos e todas as folhas. Qual é o subgrafo enraizado em g?

Exemplos Extras

Solução: O pai de c é b. Os filhos de g são h, i e j. Os irmãos de h são i e j. Os ancestrais de e são c, b e a. Os descendentes de b são c, d e e. Os vértices internos são a, b, c, g, h e j. As folhas são d, e, f, i, k, l e m. A subárvore enraizada em g está mostrada na Figura 6. ◀

As árvores enraizadas com a propriedade de que todos os vértices internos têm o mesmo número de filhos são usadas em muitas aplicações diferentes. Mais adiante, neste capítulo, usaremos essas árvores para estudar problemas que envolvem busca, ordenação e codificação.

DEFINIÇÃO 3

Links

Uma árvore enraizada é dita uma *árvore de grau m* se todo vértice interno não tiver mais que m filhos. A árvore é dita uma *árvore de grau m cheia* se cada vértice interno tiver exatamente m filhos. Uma árvore de grau m, com $m = 2$, é chamada de *árvore binária*.

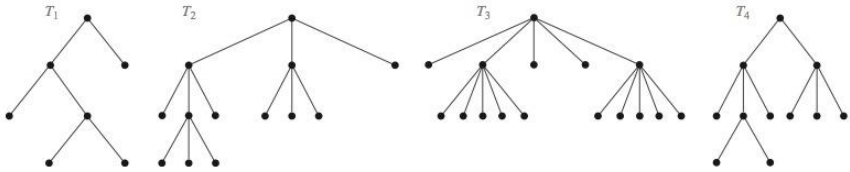


FIGURA 7 Quatro Árvores Enraizadas.

EXEMPLO 3 As árvores enraizadas na Figura 7 são árvores de grau m cheias para algum inteiro positivo m?

Solução: T_1 é uma árvore binária cheia, pois cada um de seus vértices internos tem dois filhos. T_2 é uma árvore de grau 3 cheia, pois cada um de seus vértices internos tem três filhos. Em T_3, cada vértice interno tem cinco filhos, de modo que T_3 é uma árvore de grau 5 cheia. T_4 não é uma árvore de grau m cheia para nenhum m, pois alguns de seus vértices internos têm dois e alguns têm três filhos. ◀

Uma **árvore enraizada ordenada** é uma árvore enraizada na qual os filhos de cada vértice interno estão ordenados. Árvores enraizadas ordenadas são desenhadas de modo que os filhos de cada vértice interno sejam mostrados em ordem da esquerda para a direita. Observe que uma representação de uma árvore enraizada na forma convencional determina uma ordem para suas arestas. Usaremos essas ordens das arestas ao desenhar sem mencionar explicitamente que estamos considerando uma árvore enraizada que esteja ordenada.

Em uma árvore binária ordenada (usualmente chamada apenas de **árvore binária**), se um vértice interno tiver dois filhos, o primeiro filho é chamado de **filho esquerdo** e o segundo filho é chamado de **filho direito**. A árvore enraizada no filho esquerdo de um vértice é chamada de **subárvore esquerda** deste vértice, e a árvore enraizada no filho direito de um vértice é chamada de **subárvore direita** deste vértice. O leitor deve observar que, para algumas aplicações, todo vértice de uma árvore binária, diferente da raiz, é designado como filho direito ou esquerdo de seu pai. Isso é feito mesmo quando alguns dos vértices têm apenas um filho. Faremos essas designações sempre que for necessário, mas não as faremos caso contrário.

Árvores enraizadas ordenadas podem ser definidas recursivamente. Árvores binárias, um tipo de árvores enraizadas ordenadas, foram definidas dessa forma na Seção 4.3.

EXEMPLO 4 Quais são os filhos esquerdo e direito de d na árvore binária T mostrada na Figura 8(a) (em que a ordem é a determinada pelo desenho)? Quais são as subárvores esquerda e direita de c?

Solução: O filho esquerdo de d é f e o filho direito é g. Mostramos as subárvores esquerda e direita de c nas Figuras 8(b) e 8(c), respectivamente. ◀

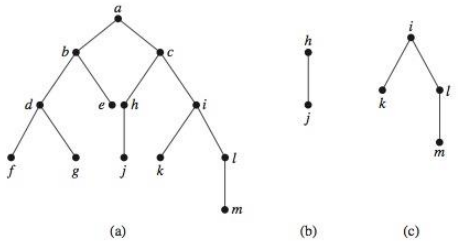


FIGURA 8 Uma Árvore Binária T e Subárvores Esquerda e Direita do Vértice c.

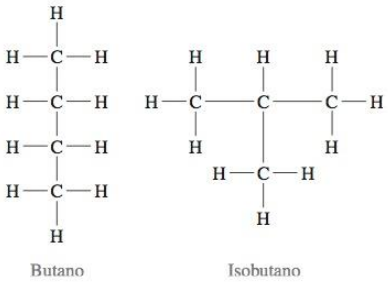


FIGURA 9 Os Dois Isômeros do Butano.

Do mesmo modo que no caso dos grafos, não existe terminologia-padrão usada para descrever árvores, árvores enraizadas, árvores enraizadas ordenadas e árvores binárias. Esta não-padronização da terminologia ocorre porque as árvores são usadas amplamente em toda a ciência da computação, a qual é um campo relativamente jovem. O leitor deve verificar cuidadosamente o significado dado aos termos que se referem a árvores sempre que eles ocorrerem.

Árvores como Modelos

As árvores são usadas como modelos em áreas tão diversas como ciência da computação, química, geologia, botânica e psicologia. Descreveremos diversos desses modelos que se baseiam em árvores.

EXEMPLO 5 **Hidrocarbonetos Saturados e Árvores** Os grafos podem ser usados para representar moléculas, nas quais os átomos são representados por vértices e as ligações entre eles pelas arestas. O matemático inglês Arthur Cayley descobriu as árvores em 1857, quando estava tentando enumerar os isômeros dos compostos da forma C_nH_{2n+2}, que são chamados de *hidrocarbonetos saturados*.

Nos modelos de grafos dos hidrocarbonetos saturados, cada átomo de carbono é representado por um vértice de grau 4 e cada átomo de hidrogênio é representado por um vértice de grau 1. Existem $3n + 2$ vértices em um grafo que representa um composto da forma C_nH_{2n+2}. O número de arestas nesse grafo é a metade da soma dos graus dos vértices. Portanto, existem $(4n + 2n + 2)/2 = 3n + 1$ arestas nesse grafo. Como esse grafo é conexo e o número de arestas é uma a menos do que o número de vértices, ele deve ser uma árvore (veja o Exercício 15 no final desta seção).

As árvores não isomorfas com n vértices de grau 4 e $2n + 2$ de grau 1 representam os isômeros diferentes de C_nH_{2n+2}. Por exemplo, quando $n = 4$, existem exatamente duas árvores não isomorfas desse tipo (o leitor deve verificar isso). Portanto, existem exatamente dois isômeros diferentes de C_4H_{10}. Suas estruturas estão mostradas na Figura 9. Estes dois isômeros são chamados de butano e isobutano. ◀

Links

ARTHUR CAYLEY (1821–1895) Arthur Cayley, filho de um mercador, mostrou o seu talento matemático muito cedo, com uma habilidade impressionante nos cálculos numéricos. Cayley ingressou no Trinity College, Cambridge, quando tinha 17 anos. Enquanto estava na faculdade, ele desenvolveu uma paixão por ler novelas. Cayley se destacou em Cambridge e foi eleito para um contrato de três anos como Fellow of Trinity e tutor-assistente. Durante essa época, Cayley começou seu estudo da geometria n-dimensional e fez diversas contribuições à geometria e à análise. Ele também desenvolveu um interesse por montanhismo, que ele aproveitava durante suas férias na Suíça. Como não havia nenhum emprego como matemático disponível para ele, Cayley deixou Cambridge e ingressou na profissão legal, sendo admitido na ordem dos advogados em 1849. Embora Cayley tenha limitado seu trabalho legal para ser capaz de continuar sua pesquisa matemática, ele ganhou uma reputação como advogado. Durante essa carreira, foi capaz de escrever mais de 300 artigos matemáticos. Em 1863, a Universidade de Cambridge abriu um novo posto em matemática e o ofereceu a Cayley. Ele aceitou o emprego, mesmo sabendo que ganharia menos do que como advogado.

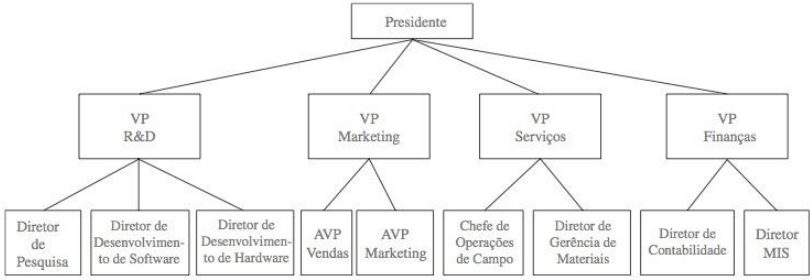


FIGURA 10 **Uma Árvore Organizacional para uma Companhia de Computadores.**

EXEMPLO 6 **Representando Organizações** A estrutura de uma grande organização pode ser modelada usando uma árvore enraizada. Cada vértice nesta árvore representa uma posição na organização. Uma aresta de um vértice a outro indica que a pessoa representada pelo vértice inicial é o chefe (direto) da pessoa representada pelo vértice final. O grafo da Figura 10 mostra uma destas árvores. Na organização representada por esta árvore, o Diretor de Desenvolvimento de Hardware trabalha diretamente para o Vice-Presidente de R&D. ◀

EXEMPLO 7 **Sistemas de Arquivo de Computador** Arquivos na memória de computador podem ser organizados em diretórios. Um diretório pode conter tanto arquivos como subdiretórios. O diretório raiz contém todo o sistema de arquivos. Logo, um sistema de arquivos pode ser representado por uma árvore enraizada, em que a raiz representa o diretório raiz, os vértices internos representam subdiretórios e folhas representam arquivos comuns ou diretórios vazios. Um sistema desse tipo é mostrado na Figura 11. Nesse sistema, o arquivo khr está no diretório rje. (Observe que links para arquivos nos quais o mesmo arquivo pode ter mais de um nome de caminho podem levar a ciclos no sistema de arquivos de computador.) ◀

EXEMPLO 8 **Processadores Paralelos Conectados em Árvore** No Exemplo 16 da Seção 9.2, descrevemos diversas redes de interconexão para processamento paralelo. Uma **rede conectada por árvore** é uma outra forma importante de interconectar processadores. O grafo que representa essa rede é

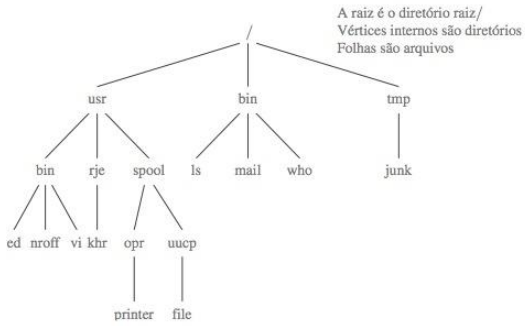


FIGURA 11 **Um Sistema de Arquivos de Computador.**

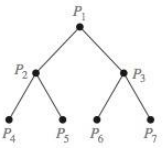


FIGURA 12 Uma Rede de Sete Processadores Conectada por Árvore.

uma árvore binária cheia. Essa rede interconecta $n = 2^k - 1$ processadores, em que k é um inteiro positivo. Um processador representado pelo vértice v, que não é uma raiz ou folha, tem três conexões de mão dupla – uma para o processador representado pelo pai de v e duas para os processadores representados pelos dois filhos de v. O processador representado pela raiz tem duas conexões de mão dupla aos processadores representados por seus dois filhos. Um processador representado por uma folha v tem uma única conexão de mão dupla ao pai de v. Mostramos uma rede conectada por árvore com sete processadores na Figura 12.

Ilustraremos como uma rede conectada por árvore pode ser usada para computação paralela. Em particular, mostraremos como os processadores na Figura 12 podem ser usados para somar oito números, usando três passos. No primeiro passo, somamos x_1 e x_2 usando P_4, x_3 e x_4 usando P_5, x_5 e x_6 usando P_6 e x_7 e x_8 usando P_7. No segundo passo, somamos $x_1 + x_2$ e $x_3 + x_4$ usando P_2 e $x_5 + x_6$ e $x_7 + x_8$ usando P_3. Finalmente, no terceiro passo, somamos $x_1 + x_2 + x_3 + x_4$ e $x_5 + x_6 + x_7 + x_8$ usando P_1. Os três passos usados para somar oito números se comparam favoravelmente aos sete passos necessários para somar oito números de modo serial, em que os passos são a adição de um número à soma dos números anteriores na lista. ◀

Propriedades das Árvores

Freqüentemente, precisaremos de resultados que relacionem o número de arestas e vértices de diversos tipos em árvores.

TEOREMA 2 Uma árvore com n vértices tem $n - 1$ arestas.

Links

Demonstração: Usaremos indução matemática para demonstrar este teorema. Observe que para todas as árvores aqui, podemos escolher uma raiz e considerar a árvore com raiz.

PASSO–BASE: Quando $n = 1$, uma árvore com $n = 1$ vértice não tem nenhuma aresta. Segue que o teorema é verdadeiro para $n = 1$.

PASSO DE INDUÇÃO: A hipótese de indução afirma que toda árvore com k vértices tem $k - 1$ arestas, em que k é um inteiro positivo. Suponha que uma árvore T tenha $k + 1$ vértices e que v seja uma folha de T (que deve existir, visto que a árvore é finita) e w seja o pai de v. A remoção de T do vértice v e da aresta que liga w a v produz uma árvore T' com k vértices, pois o grafo resultante ainda é conexo e não tem nenhum ciclo simples. Pela hipótese de indução, T' tem $k - 1$ arestas. Segue que T tem k arestas, pois ela tem uma aresta a mais do que T', a aresta que liga v a w. Isso completa o passo de indução. ◁

O número de vértices em uma árvore de grau m cheia com um número especificado de vértices internos é determinado, como mostra o Teorema 3. Como no Teorema 2, usaremos n para indicar o número de vértices em uma árvore.

TEOREMA 3 Uma árvore de grau m cheia com i vértices internos contém $n = mi + 1$ vértices.

Demonstração: Todo vértice, exceto a raiz, é o filho de um vértice interno. Como cada um dos vértices internos i tem m filhos, existem mi vértices na árvore além da raiz. Portanto, a árvore contém $n = mi + 1$ vértices. ◁

Suponha que T seja uma árvore de grau m cheia. Seja i o número de vértices internos e l o número de folhas nesta árvore. Uma vez que um entre n, i e l seja conhecido, as outras duas quantidades estão determinadas. O Teorema 4 explica como encontrar as outras duas quantidades a partir da que é conhecida.

TEOREMA 4 Uma árvore de grau m cheia com

(i) n vértices tem $i = (n-1)/m$ vértices internos e $l = [(m-1)n+1]/m$ folhas,
(ii) i vértices internos tem $n = mi+1$ vértices e $l = (m-1)i+1$ folhas,
(iii) l folhas tem $n = (ml-1)/(m-1)$ vértices e $i = (l-1)/(m-1)$ vértices internos.

Demonstração: Represente o número de vértices por n, o número de vértices internos por i e o número de folhas por l. As três partes do teorema podem ser demonstradas usando a igualdade dada no Teorema 3, isto é, $n = mi+1$, junto com a igualdade $n = l+i$, que é verdadeira porque cada vértice ou é uma folha ou é um vértice interno. Demonstraremos aqui a parte (i). As demonstrações das partes (ii) e (iii) são deixadas como exercícios para o leitor.

Isolando i em $n = mi+1$ obtemos $i = (n-1)/m$. Então, a substituição dessa expressão para i na equação $n = l+i$ mostra que $l = n-i = n-(n-1)/m = [(m-1)n+1]/m$. ◁

O Exemplo 9 ilustra como o Teorema 4 pode ser usado.

EXEMPLO 9 Suponha que alguém inicie uma corrente de cartas. É pedido a cada pessoa que recebe a carta que a mande para quatro outras pessoas. Algumas pessoas fazem isso, mas outras não mandam nenhuma carta. Quantas pessoas viram a carta, incluindo a primeira pessoa, se ninguém recebeu mais de uma carta e se a corrente termina depois de ter havido 100 pessoas que leram, mas não mandaram a carta? Quantas pessoas mandaram a carta?

Solução: A corrente de cartas pode ser representada usando uma árvore de grau 4. Os vértices internos correspondem a pessoas que mandaram a carta, e as folhas correspondem a pessoas que não a mandaram. Como 100 pessoas não mandaram a carta, o número de folhas nesta árvore enraizada é $l = 100$. Logo, a parte (iii) do Teorema 4 mostra que o número de pessoas que viram a carta é $n = (4 \cdot 100 - 1)/(4-1) = 133$. Além disso, o número de vértices internos é $133 - 100 = 33$, então 33 pessoas mandaram a carta. ◀

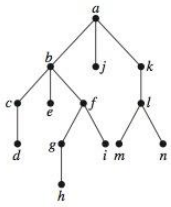


FIGURA 13 Uma Árvore Enraizada.

Em geral, é desejável usar árvores enraizadas que sejam "balanceadas", de modo que as subárvores em cada vértice contenham caminhos de aproximadamente o mesmo comprimento. Algumas definições tornarão este conceito claro. O **nível** de um vértice v em uma árvore enraizada é o comprimento do único caminho da raiz a este vértice. O nível da raiz é definido como zero. A **altura** de uma árvore enraizada é o máximo dos níveis dos vértices. Em outras palavras, a altura de uma árvore enraizada é o comprimento do caminho mais longo da raiz a qualquer vértice.

EXEMPLO 10 Encontre o nível de cada vértice da árvore enraizada mostrada na Figura 13. Qual é a altura desta árvore?

Solução: A raiz a está no nível 0. Os vértices b, j e k estão no nível 1. Os vértices c, e, f e l estão no nível 2. Os vértices d, g, i, m e n estão no nível 3. Finalmente, o vértice h está no nível 4. Como o maior nível de qualquer vértice é 4, esta árvore tem altura 4. ◀

Uma árvore de grau m enraizada de altura h é **balanceada** se todas as folhas estiverem nos níveis h ou $h-1$.

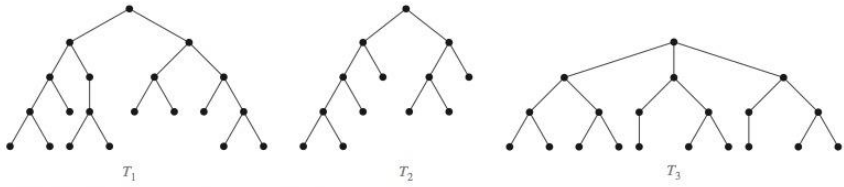


FIGURA 14 **Algumas Árvore Enraizadas.**

EXEMPLO 11 Quais das árvores enraizadas mostradas na Figura 14 são balanceadas?

Solução: T_1 é balanceada, pois todas as suas folhas estão nos níveis 3 e 4. Entretanto, T_2 não é balanceada, pois tem folhas nos níveis 2, 3 e 4. Finalmente, T_3 é balanceada, pois todas as suas folhas estão no nível 3. ◀

O resultado do Teorema 5 relaciona a altura e o número de folhas em árvores de grau m.

TEOREMA 5 Existem no máximo m^h folhas em uma árvore de grau m de altura h.

Demonstração: A demonstração usa indução matemática na altura. Primeiro, considere árvores de grau m de altura 1. Estas árvores consistem de uma raiz com não mais que m filhos, cada um como uma folha. Logo, não existem mais de $m^1 = m$ folhas em uma árvore de grau m de altura 1. Este é o passo-base do argumento de indução.

Suponha agora que o resultado seja válido para todas as árvores de grau m de altura menor que h; esta é a hipótese de indução. Seja T uma árvore de grau m de altura h. As folhas de T são as folhas das subárvores de T obtidas, removendo as arestas da raiz a cada um dos vértices de nível 1, como mostrado na Figura 15.

Cada uma destas subárvores tem altura menor que ou igual a $h - 1$. Assim, pela hipótese de indução, cada uma destas árvores enraizadas tem, no máximo, m^{h-1} folhas. Como existem, no máximo, m destas subárvores, cada uma com, no máximo, m^{h-1} folhas, existem, no máximo, $m \cdot m^{h-1} = m^h$ folhas na árvore enraizada. Isso completa o argumento. ◁

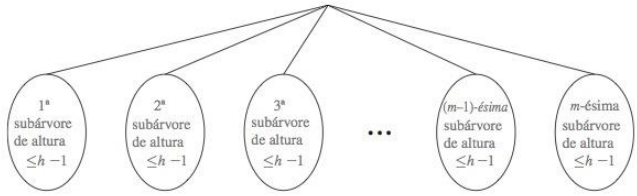


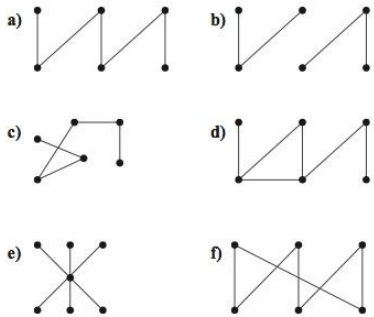
FIGURA 15 **O Passo de Indução da Demonstração.**

COROLÁRIO 1 Se uma árvore de grau m de altura h tem l folhas, então $h \geq \lceil \log_m l \rceil$. Se a árvore de grau m é cheia e balanceada, então $h = \lceil \log_m l \rceil$. (Estamos usando aqui a função menor inteiro maior que. Lembre que $\lceil x \rceil$ é o menor inteiro maior que ou igual a x.)

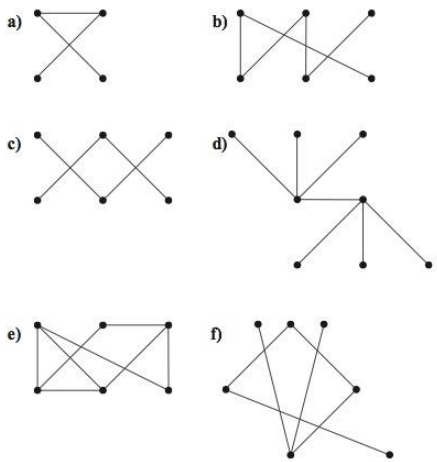
Demonstração: Sabemos, pelo Teorema 5, que $l \leq m^h$. Tomando os logaritmos na base m, obtemos $\log_m l \leq h$. Como h é um inteiro, temos $h \geq \lceil \log_m l \rceil$. Suponha agora que a árvore seja balanceada. Então, cada folha está no nível h ou $h - 1$, e como a altura é h, existe pelo menos uma folha no nível h. Segue que deve haver mais do que m^{h-1} folhas (veja o Exercício 30 no final desta seção.) Como $l \leq m^h$, temos que $m^{h-1} < l \leq m^h$. Tomando logaritmos na base m nesta desigualdade, obtemos $h - 1 < \log_m l \leq h$. Portanto, $h = \lceil \log_m l \rceil$. ◁

Exercícios

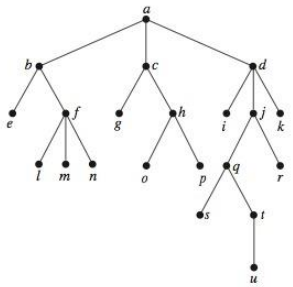
1. Quais destes grafos são árvores?



2. Quais destes grafos são árvores?

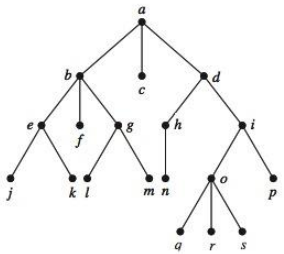


3. Responda estas questões sobre a árvore enraizada ilustrada.



a) Qual vértice é a raiz?
b) Quais vértices são internos?
c) Quais vértices são folhas?
d) Quais vértices são filhos de j?
e) Qual vértice é o pai de h?
f) Quais vértices são irmãos de o?
g) Quais vértices são ancestrais de m?
h) Quais vértices são descendentes de b?

4. Responda às mesmas questões listadas no Exercício 3 para a árvore enraizada a seguir.



5. A árvore enraizada do Exercício 3 é uma árvore de grau m cheia para algum inteiro positivo m?

6. A árvore enraizada do Exercício 4 é uma árvore de grau m cheia para algum inteiro positivo m?

7. Qual é o nível de cada vértice da árvore enraizada do Exercício 3?

8. Qual é o nível de cada vértice da árvore enraizada do Exercício 4?

9. Desenhe subárvore da árvore do Exercício 3 que tenha raiz em

a) a. **b)** c. **c)** e.

10. Desenhe subárvore da árvore do Exercício 4 que tenha raiz em

a) a. **b)** c. **c)** e.

11. **a)** Quantas árvores com três vértices, sem raiz, não isomorfas, existem?

b) Quantas árvores com três vértices, com raiz, não isomorfas, existem (usando isomorfismos para grafos orientados)?

***12.** **a)** Quantas árvores com quatro vértices, sem raiz, não isomorfas, existem?

b) Quantas árvores com quatro vértices, com raiz, não isomorfas, existem (usando isomorfismos para grafos orientados)?

***13.** **a)** Quantas árvores com cinco vértices, sem raiz, não isomorfas, existem?

b) Quantas árvores com cinco vértices, com raiz, não isomorfas, existem (usando isomorfismos para grafos orientados)?

***14.** Mostre que um grafo simples sem ciclos é uma árvore se e somente se ele é conexo, mas a remoção de qualquer uma de suas arestas produz um grafo que não é conexo.

☞ ***15.** Seja G um grafo simples com n vértices. Mostre que G é uma árvore se e somente se G for conexo e tiver $n - 1$ arestas. [*Dica:* Para a parte "se", use o Exercício 14 e o Teorema 2.]

16. Quais grafos bipartidos completos $K_{m,n}$, em que m e n são inteiros positivos, são árvores?

17. Quantas arestas uma árvore com 10.000 vértices deve ter?

18. Quantos vértices uma árvore de grau 5 cheia com 100 vértices internos deve ter?

19. Quantas arestas uma árvore binária cheia com 1000 vértices internos deve ter?

20. Quantas folhas uma árvore de grau 3 cheia com 100 vértices deve ter?

21. Suponha que 1000 pessoas entrem em um torneio de xadrez. Use um modelo de árvore com raiz de torneio para determinar quantas partidas devem ser jogadas para determinar o campeão, se um jogador é eliminado depois de perder uma partida e os jogos continuam até que apenas um participante não tenha perdido. (Suponha que não haja empates.)

22. Uma corrente de cartas começa quando uma pessoa manda uma carta para cinco outras pessoas. Cada pessoa que recebe a carta ou a manda para cinco outras pessoas que nunca a receberam ou não a manda para ninguém. Suponha que 10.0000 pessoas mandem a carta antes de a corrente terminar e que ninguém receba mais de uma carta. Quantas pessoas receberam a carta e quantas não a mandaram?

23. Uma corrente de cartas começa quando uma pessoa manda uma carta para outras 10 pessoas. É pedido a cada pessoa que mande a carta para 10 outras, e que cada carta tenha uma lista das seis pessoas anteriores na corrente. A menos que haja menos de seis pessoas na lista, cada pessoa manda um dólar para a primeira pessoa nesta lista, remove o nome desta pessoa da lista, move o nome de cada uma das outras cinco pessoas uma posição para cima e acrescenta seu nome no final da lista. Se nenhuma pessoa quebrar a corrente e ninguém receber mais de uma carta, quanto dinheiro uma pessoa na corrente acabará recebendo?

***24.** Ou desenhe uma árvore de grau m cheia com 76 folhas e altura 3, em que m é um intero positivo, ou mostre que essa árvore não existe.

***25.** Ou desenhe uma árvore de grau m cheia com 84 folhas e altura 3, em que m é um inteiro positivo, ou mostre que essa árvore não existe.

***26.** Uma árvore de grau m cheia T tem 81 folhas e altura 4.

a) Dê limitantes superior e inferior para m.

b) O que será m se, além disso, T for balanceada?

Uma **árvore de grau m completa** é uma árvore de grau m cheia, na qual toda folha está no mesmo nível.

27. Construa uma árvore binária completa de altura 4 e uma árvore de grau 3 completa de altura 3.

28. Quantos vértices e quantas folhas tem uma árvore de grau m completa de altura h?

29. Demonstre

a) a parte (*ii*) do Teorema 4.

b) a parte (*iii*) do Teorema 4.

☞ **30.** Mostre que uma árvore balanceada de grau m cheia de altura h tem mais de m^{h-1} folhas.

31. Quantas arestas existem em uma floresta de t árvores que contém um total de n vértices?

32. Explique como uma árvore pode ser usada para representar a tabela do conteúdo de um livro organizado em capítulos, em que cada capítulo é organizado em seções e cada seção é organizada em subseções.

33. Quantos isômeros diferentes têm estes hidrocarbonetos saturados?

a) C_3H_8 **b)** C_5H_{12} **c)** C_6H_{14}

34. O que cada um destes representa em uma árvore organizacional?

a) o pai de um vértice.

b) um filho de um vértice.

c) um irmão de um vértice.

d) os ancestrais de um vértice.

e) os descendentes de um vértice.

f) o nível de um vértice.

g) a altura da árvore.

35. Responda as mesmas questões dadas no Exercício 34 para uma árvore enraizada que representa um sistema de arquivos de computador.

36. **a)** Desenhe uma árvore binária completa com 15 vértices que representa uma rede conectada por árvore de 15 processadores.

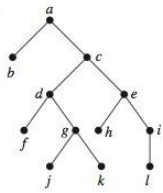
b) Mostre como 16 números podem ser somados usando os 15 processadores da parte (a) em quatro passos.

37. Seja n uma potência de 2. Mostre que n números podem ser somados em $\log n$ passos usando uma rede conectada por árvore com $n - 1$ processadores.

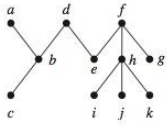
***38.** Uma **árvore rotulada** é uma árvore na qual foi associado um rótulo para cada vértice. Duas árvores rotuladas são consideradas isomorfas quando existe um isomorfismo entre elas que preserva os rótulos dos vértices. Quantas árvores não isomorfas existem com três vértices rotulados com inteiros diferentes do conjunto $\{1, 2, 3\}$? Quantas árvores não isomorfas existem com quatro vértices rotulados com inteiros diferentes do conjunto $\{1, 2, 3, 4\}$?

A **excentricidade** de um vértice em uma árvore sem raiz é o comprimento do caminho simples mais longo que começa neste vértice. Um vértice é chamado de **centro** se nenhum vértice na árvore tiver excentricidade menor do que este vértice. Nos exercícios 39 a 41, encontre todo vértice que seja um centro na árvore dada.

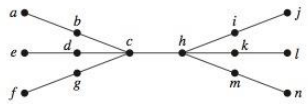
39.



40.



41.



42. Mostre que deveria ser escolhido um centro como raiz para produzir uma árvore enraizada de altura mínima a partir da árvore sem raiz.

***43.** Mostre que uma árvore ou tem um único centro ou tem dois centros que são adjacentes.

44. Mostre que toda árvore pode ser colorida usando duas cores.

As **árvores de Fibonacci enraizadas** T_n são definidas recursivamente da seguinte maneira. T_1 e T_2 são ambas árvores enraizadas que consistem de um único vértice, e para $n = 3, 4, \ldots$, a árvore enraizada T_n é construída a partir de uma raiz com T_{n-1} como sua subárvore esquerda e T_{n-2} como sua subárvore direita.

45. Desenhe as primeiras sete árvores de Fibonacci enraizadas.

***46.** Quantos vértices, folhas e vértices internos a árvore de Fibonacci enraizada T_n tem, em que n é um inteiro positivo? Qual é a sua altura?

47. O que está errado na seguinte "demonstração", usando indução matemática, da afirmação que toda árvore com n vértices tem um caminho simples de comprimento $n - 1$. *Passo-base*: Toda árvore com um vértice tem um caminho de comprimento 0. *Passo de indução:* Suponha que uma árvore com n vértices tenha um caminho simples de comprimento $n - 1$, que tem u como seu vértice final. Acrescente um vértice v e a aresta de u a v. A árvore resultante tem $n + 1$ vértices e um caminho simples de comprimento n. Isso completa o passo de indução.

☞***48.** Mostre que a profundidade média de uma folha em uma árvore binária com n vértices é $\Omega(\log n)$.

10.2 Aplicações de Árvores

Introdução

Discutiremos três problemas que podem ser estudados usando árvores. O primeiro problema é: como os itens em uma lista devem ser armazenados de modo que um item possa ser facilmente localizado? O segundo problema é: que série de decisões deveria ser tomada para encontrar um objeto com uma certa propriedade em uma coleção de objetos de um certo tipo? O terceiro problema é: como um conjunto de caracteres poderia ser eficientemente codificado por seqüências de bits?

Árvores de Busca Binária

Links

A procura de itens em uma lista é uma das tarefas mais importantes que aparecem na ciência da computação. Nosso objetivo primário é implementar um algoritmo de busca que encontre itens eficientemente quando os itens estão totalmente ordenados. Pode-se conseguir isso através do uso

de uma **árvore de busca binária**, que é uma árvore binária na qual cada filho de um vértice é designado como um filho direito ou esquerdo, nenhum vértice tem mais do que um filho direito ou esquerdo e cada vértice é rotulado com uma chave, que é um dos itens. Além disso, são associadas chaves aos vértices, de modo que a chave de um vértice é tanto maior que as chaves de todos os vértices em sua subárvore esquerda quanto menor que as chaves de todos os vértices em sua subárvore direita.

Este procedimento recursivo é usado para formar a árvore de busca binária para uma lista de itens. Comece com uma árvore que contenha somente um vértice, ou seja, a raiz. O primeiro item nesta lista é associado como a chave da raiz. Para adicionar um novo item, primeiro compare-o com as chaves dos vértices que já estão na árvore, começando na raiz e se movendo para a esquerda se o item for menor que a chave do vértice respectivo se este vértice tiver um filho esquerdo, ou, se movendo para a direita se o item for maior que a chave do vértice respectivo se este vértice tiver um filho direito. Quando o item for menor que o vértice respectivo e este vértice não tiver filho esquerdo, então um novo vértice com este item como sua chave é inserido como um novo filho esquerdo. Analogamente, quando o item for maior que o vértice respectivo e este vértice não tiver filho direito, então um novo vértice com este item como sua chave é inserido como um novo filho direito. Ilustramos este procedimento com o Exemplo 1.

EXEMPLO 1 Forme uma árvore de busca binária para as palavras *mathematics, physics, geography, zoology, meteorology, geology, psychology* e *chemistry* (usando a ordem alfabética).

Solução: A Figura 1 mostra os passos usados para construir esta árvore de busca binária. A palavra *mathematics* é a chave da raiz. Como *physics* vem depois de *mathematics* (em ordem alfabética), adicione um filho direito à raiz com a chave *physics*. Como *geography* vem antes de *mathematics*, adicione um filho esquerdo à raiz com a chave *geography*. A seguir, adicione um filho direito ao vértice com a chave *physics* e associe-o à chave *zoology*, pois *zoology* vem depois de *mathematics* e depois de *physics*. Analogamente, adicione um filho esquerdo ao vértice com a chave *physics* e associe a este novo vértice a chave *meteorology*. Adicione um filho direito ao vértice com a chave *geography* e associe este novo vértice à chave *geology*. Adicione um filho esquerdo ao vértice com a chave *zoology* e associe-o à chave *psychology*.

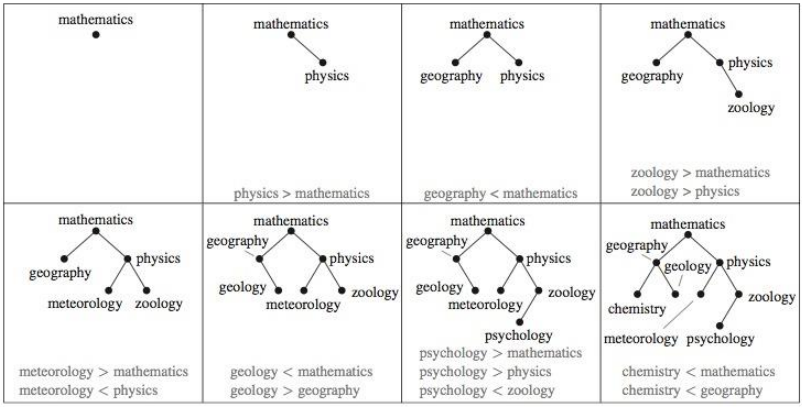


FIGURA 1 A Construção de uma Árvore de Busca Binária.

Adicione um filho esquerdo ao vértice com a chave *geography* e associe-o à chave *chemistry*. (O leitor deve fazer todas as comparações necessárias em cada passo.) ◀

Uma vez que tenhamos uma árvore de busca binária, precisamos de uma maneira de localizar itens nessa árvore, bem como uma maneira de adicionar novos itens. O Algoritmo 1, um algoritmo de inserção, na realidade executa ambas as tarefas, mesmo que possa parecer que ele é projetado apenas para adicionar vértices a uma árvore de busca binária. Ou seja, o Algoritmo 1 é um procedimento que localiza um item x em uma árvore de busca binária se ele estiver presente, e adiciona um novo vértice com x como sua chave se x não estiver presente. No pseudocódigo, v é o vértice que está sendo examinado no momento e $label(v)$ representa a chave deste vértice. O algoritmo começa pelo exame da raiz. Se x for igual à chave de v, então o algoritmo encontrou a posição de x e pára; se x for menor que a chave de v, nos movemos para o filho esquerdo de v e repetimos o procedimento; e se x for maior que a chave de v, nos movemos para o filho direito de v e repetimos o procedimento. Se em qualquer passo tentarmos nos mover para um filho que não está presente, saberemos que x não está presente na árvore e adicionamos um novo vértice como este filho com x como sua chave.

ALGORITMO 1 Localização e Acréscimo de Itens a Uma Árvore de Busca Binária.

procedure *insercao*(T: árvore de busca binária, x: item)
v := raiz de T
(um vértice que não esteja presente em T tem o valor *nulo*)
while $v \neq$ *nulo* e $label(v) \neq x$
begin
  if $x < label(v)$ **then**
    if filho esquerdo de $v \neq$ *nulo* **then** v := filho esquerdo de v
    else acrescente *novo vértice* como um filho esquerdo de v e faça $v :=$ *nulo*
  else
    if filho direito de $v \neq$ *nulo* **then** v := filho direito de v
    else acrescente *novo vértice* como um filho direito de v a T e faça $v :=$ *nulo*
end
if raiz de $T =$ *nulo* **then** acrescente um vértice v à árvore e rotule-o com x
else if v for *nulo* ou $label(v) \neq x$ **then** rotule *novo vértice* com x e considere v como este novo vértice
{v = posição de x}

O Exemplo 2 ilustra o uso do Algoritmo 1 para inserir um novo item em uma árvore de busca binária.

EXEMPLO 2 Use o Algoritmo 1 para inserir a palavra *oceanography* na árvore de busca binária do Exemplo 1.

Solução: O Algoritmo 1 começa com v, o vértice que está sendo examinado, sendo a raiz de T, de modo que $label(v) = mathematics$. Como $v \neq nulo$ e $label(v) = mathematics < oceanography$, examinamos a seguir o filho direito da raiz. Este filho direito existe, de modo que consideramos v, o vértice que está sendo examinado, como este filho direito. Neste passo, temos $v \neq nulo$ e $label(v) = physics > oceanography$, de modo que examinamos o filho esquerdo de v. Este filho esquerdo existe, de modo que consideramos v, o vértice que está sendo examinado, como este filho esquerdo. Neste passo, também temos $v \neq nulo$ e $label(v) = metereology < oceanography$, de modo que tentamos examinar o filho direito de v. Entretanto, este filho direito não existe, de modo que adicionamos um novo vértice como filho direito de v (o qual, neste ponto, é o vértice com a chave *metereology*) e consideramos $v := nulo$. Agora, saímos do laço do **while** porque $v = nulo$. Como a raiz de T não é *nula* e $v = nulo$, usamos a instrução **else if** no final do algoritmo para rotular nosso novo vértice com a chave *oceanography*. ◀

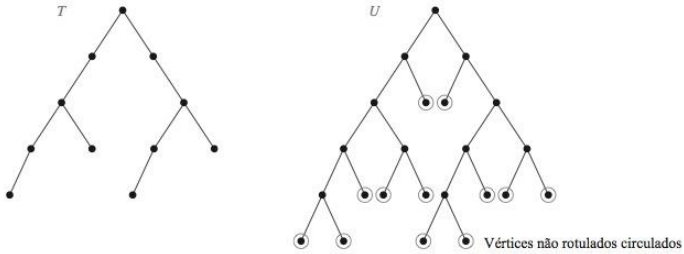


FIGURA 2 Adição de Vértices Não Rotulados para Tornar uma Árvore de Busca Binária Cheia.

Determinaremos agora a complexidade computacional deste procedimento. Suponha que tenhamos uma árvore de busca binária T para uma lista de n itens. Podemos formar uma árvore binária cheia U a partir de T, adicionando vértices não rotulados sempre que necessário, de modo que todo vértice com uma chave tenha dois filhos. Isto é ilustrado na Figura 2. Uma vez que tenhamos feito isso, podemos facilmente localizar ou adicionar um novo item como uma chave sem adicionar um vértice.

O máximo de comparações necessárias para adicionar um novo item é o comprimento do caminho mais longo em U da raiz a uma folha. Os vértices internos de U são os vértices de T. Segue que U tem n vértices internos. Podemos agora usar a parte (*ii*) do Teorema 4 na Seção 10.1 para concluir que U tem $n + 1$ folhas. Usando o Corolário 1 da Seção 10.1, vemos que a altura de U é maior que ou igual a $h = \lceil \log(n + 1) \rceil$. Conseqüentemente, é necessário executar pelo menos $\lceil \log(n + 1) \rceil$ comparações para adicionar algum item. Observe que se U for balanceada, sua altura é $\lceil \log(n + 1) \rceil$ (pelo Corolário 1 da Seção 10.1). Logo, se uma árvore de busca binária for balanceada, localizar ou adicionar um item não requer mais do que $\lceil \log(n + 1) \rceil$ comparações. Uma árvore de busca binária pode se tornar não balanceada conforme forem adicionados itens a ela. Como árvores de busca binária balanceadas dão uma complexidade de pior caso ótima para a busca binária, foram desenvolvidos algoritmos que rebalanceiam árvores de busca binária conforme são adicionados itens. O leitor interessado pode consultar as referências sobre estrutura de dados para a descrição desses algoritmos.

Árvores de Decisão

Links

Árvores enraizadas podem ser usadas para modelar problemas nos quais uma série de decisões leva a uma solução. Por exemplo, uma árvore de busca binária pode ser usada para localizar itens com base em uma série de comparações, em que cada comparação nos diz se já localizamos o item ou se devemos ir para a direita ou para a esquerda em uma subárvore. Uma árvore com raiz na qual cada vértice interno corresponde a uma decisão, com uma subárvore nestes vértices para cada saída possível da decisão, é chamada **árvore de decisão**. As soluções possíveis do problema correspondem aos caminhos para as folhas desta árvore enraizada. O Exemplo 3 ilustra uma aplicação de árvores de decisão.

EXEMPLO 3 Suponha que existam sete moedas, todas com o mesmo peso, e uma moeda falsa com o peso menor que o das outras. Quantas pesagens são necessárias, usando uma balança de pratos, para determinar qual das oito moedas é a falsa? Dê um algoritmo para encontrar esta moeda falsa.

Exemplos Extras

Solução: Há três possibilidades para cada pesagem em uma balança de pratos. Os dois pratos podem ter alturas iguais, o primeiro pode ser mais pesado, ou o segundo pode ser mais pesado. Conseqüentemente, a árvore de decisão para a seqüência de pesagem é uma árvore de grau 3.

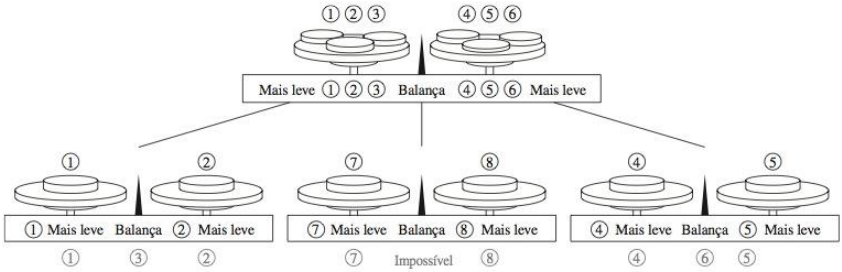


FIGURA 3 **Uma Árvore de Decisão para Localizar uma Moeda Falsa. A moeda falsa está mostrada em cinza abaixo de cada pesagem final.**

Existem pelo menos oito folhas na árvore de decisão, pois existem oito saídas possíveis (e cada uma das oito moedas pode ser a moeda falsa mais leve) e cada saída possível deve ser representada por, pelo menos, uma folha. O maior número de pesagens necessárias para determinar a moeda falsa é a altura da árvore de decisão. Do Corolário 1 da Seção 10.1 segue que a altura da árvore de decisão é, pelo menos, $\lceil \log_3 8 \rceil = 2$. Portanto, serão necessárias pelo menos duas pesagens.

É possível determinar a moeda falsa usando duas pesagens. A árvore de decisão que ilustra como isso é feito é mostrada na Figura 3. ◀

A COMPLEXIDADE DOS ALGORITMOS DE ORDENAÇÃO Muitos algoritmos de ordenação diferente foram desenvolvidos. Para decidir se um algoritmo de ordenação particular é eficiente, deve-se determinar sua complexidade. Usando árvores de decisão como modelos, pode ser encontrado um limitante inferior para a complexidade de pior caso dos algoritmos de ordenação.

Podemos usar árvores de decisão para modelar algoritmos de ordenação e para determinar uma estimativa para a complexidade de pior caso destes algoritmos. Observe que dados n elementos, existem $n!$ ordenações possíveis destes elementos, pois cada uma das $n!$ permutações destes elementos pode ser a ordem correta. Os algoritmos de ordenação estudados neste livro, e os algoritmos de ordenação mais comumente usados, baseiam-se em comparações binárias, ou seja, na comparação de dois elementos de cada vez. O resultado de cada uma destas comparações estreita o conjunto das ordenações possíveis. Assim, um algoritmo de ordenação que se baseia em comparações binárias pode ser representado por uma árvore de decisão binária, na qual cada vértice interno representa uma comparação de dois elementos. Cada folha representa uma das $n!$ permutações de n elementos.

EXEMPLO 4 Mostramos na Figura 4 uma árvore de decisão que ordena os elementos da lista a, b e c. ◀

A complexidade de uma ordenação que se baseia em comparações binárias é medida em termos do número dessas comparações que são usadas. O maior número de comparações binárias necessárias para ordenar uma lista com n elementos fornece o desempenho de pior caso deste algoritmo. O máximo de comparações usadas é igual ao comprimento do caminho mais comprido na árvore de decisão que representa o procedimento de ordenação. Em outras palavras, o maior número de comparações jamais necessário é igual à altura da árvore de decisão. Como a altura de uma árvore binária com $n!$ folhas é, pelo menos, $\lceil \log n! \rceil$ (usando o Corolário 1 da Seção 10.1), pelo menos $\lceil \log n! \rceil$ comparações serão necessárias, como afirma o Teorema 1.

TEOREMA 1 Um algoritmo de ordenação que se baseia em comparações binárias exige, pelo menos, $\lceil \log n! \rceil$ comparações.

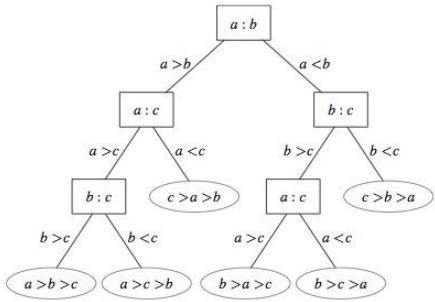


FIGURA 4 **Uma Árvore de Decisão para Ordenar Três Elementos Distintos.**

Podemos usar o Teorema 1 para fornecer uma estimativa Ômega grande para o número de comparações usadas por um algoritmo de ordenação que se baseia em comparações binárias. Precisamos apenas observar que pelo Exercício 62 da Seção 3.2 sabemos que $\lceil \log n! \rceil$ é $\Theta(n \log n)$, uma das funções de referência mais comumente usadas na complexidade computacional de algoritmos. O Corolário 1 é uma conseqüência desta estimativa.

COROLÁRIO 1 O número de comparações usado por um algoritmo de ordenação para ordenar n elementos com base em comparações binárias é $\Omega(n \log n)$.

Uma conseqüência do Corolário 1 é que um algoritmo de ordenação que se baseia em comparações binárias que usa $\Theta(n \log n)$ comparações, no pior caso, para ordenar n elementos é ótimo, no sentido de que nenhum outro algoritmo tem melhor complexidade de pior caso. Observe que pelo Teorema 1 da Seção 4.4, vemos que o algoritmo de merge sort é ótimo neste sentido.

Podemos também estabelecer um resultado análogo para a complexidade de caso médio dos algoritmos de ordenação. O número médio de comparações usadas por um algoritmo de ordenação que se baseia em comparações binárias é a profundidade média de uma folha na árvore de decisão que representa o algoritmo de ordenação. Pelo Exercício 48 da Seção 10.1, sabemos que a profundidade média de uma folha em uma árvore binária com N vértices é $\Omega(\log N)$. Obtemos a seguinte estimativa quando tomamos $N = n!$ e observamos que uma função que é $\Omega(\log n!)$ também é $\Omega(n \log n)$, pois $n!$ é $\Theta(n \log n)$.

TEOREMA 2 O número médio de comparações usadas por um algoritmo de ordenação para ordenar n elementos que se baseiam em comparações binárias é $\Omega(n \log n)$.

Códigos de Prefixo

Considere o problema de usar seqüências de bits para codificar as letras do alfabeto da língua inglesa (em que não é feita nenhuma distinção entre letras maiúsculas e minúsculas). Podemos representar cada letra por uma seqüência de bits de comprimento cinco, pois existem 26 letras e 32 seqüências de bits de comprimento cinco. O número total de bits usados para codificar dados é cinco vezes maior que o número de caracteres no texto, quando cada caractere for codificado com cinco bits. É possível encontrar um sistema de codificação destas letras tal que, quando da-

dos forem codificados, sejam usados menos bits? Podemos economizar memória e reduzir o tempo de transmissão se isso puder ser feito.

Considere o uso de seqüências de bits de comprimentos diferentes para codificar letras. As letras que ocorrem mais freqüentemente deveriam ser codificadas usando seqüências de bits mais curtas, e as seqüências de bits mais longas deveriam ser usadas para codificar letras que ocorrem raramente. Quando letras são codificadas usando números variáveis de bits, deve ser usado algum método para determinar onde os bits para cada caractere começam e terminam. Por exemplo, se *e* for codificado por 0, *a* por 1 e *t* por 01, então a seqüência de bits 0101 poderia corresponder a *eat*, *tea*, *eaea* ou *tt*.

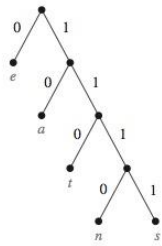


FIGURA 5 A Árvore Binária com um Código de Prefixo.

Uma maneira de garantir que nenhuma seqüência de bits corresponda a mais de uma seqüência de letras é codificar as letras de modo que a seqüência de bits para uma letra nunca ocorra como a primeira parte de uma seqüência de bits para uma outra letra. Códigos com esta propriedade são chamados de **códigos de prefixo**. Por exemplo, a codificação de *e* como 0, *a* como 10 e *t* como 11 é um código de prefixo. Uma palavra pode ser recuperada a partir da única seqüência de bits que codifica suas letras. Por exemplo, a seqüência 10110 é a codificação de *ate*. Para ver isso, observe que o 1 inicial não representa um caractere, mas 10 representa *a* (e não poderia ser a primeira parte de uma seqüência de bits de outra letra). A seguir, o próximo 1 não representa um caractere, mas 11 representa a letra *t*. O bit final, 0, representa *e*.

Um código de prefixo pode ser representado usando uma árvore binária, em que os caracteres são os rótulos das folhas na árvore. As arestas da árvore são rotuladas de modo que a uma aresta que leva a um filho esquerdo seja associado 0 e a uma aresta que leva a um filho direito seja associado 1. A seqüência de bits usada para codificar um caractere é a seqüência de rótulos das arestas do único caminho da raiz à folha que tem este caractere como seu rótulo. Por exemplo, a árvore na Figura 5 representa a codificação de *e* por 0, *a* por 10, *t* por 110, *n* por 1110 e *s* por 1111.

A árvore que representa um código pode ser usada para decodificar uma seqüência de bits. Por exemplo, considere a palavra codificada por 11111011100 usando o código da Figura 5. Esta seqüência de bits pode ser decodificada começando na raiz, usando a seqüência de bits para formar um caminho que termina quando uma folha é atingida. Cada bit 0 dirige o caminho pela aresta que leva ao filho esquerdo do último vértice do caminho, e cada bit 1 corresponde ao filho direito deste vértice. Conseqüentemente, o 1111 inicial corresponde ao caminho que começa na raiz, vai para a direita quatro vezes, levando a uma folha no grafo que tem *s* como seu rótulo, pois a seqüência 1111 é o código para *s*. Continuando com o quinto bit, atingimos uma folha depois de ir à direita e, a seguir, à esquerda, quando o vértice rotulado com *a*, que é codificado por 10, é visitado. Começando com o sétimo bit, atingimos uma folha depois de ir para a direita três vezes e, a seguir, para a esquerda, quando o vértice rotulado com *n*, que é codificado por 1110, é visitado. Finalmente, o último bit, 0, leva à folha que é rotulada com *e*. Portanto, a palavra original é *sane*.

Podemos construir um código de prefixo a partir de qualquer árvore binária na qual a aresta esquerda de cada vértice interno está rotulada por 0 e a aresta direita por 1, e na qual as folhas estão rotuladas pelos caracteres. Cada caractere é codificado pela seqüência de bits construída usando os rótulos das arestas no único caminho da raiz às folhas.

Links CODIFICAÇÃO DE HUFFMAN Introduzimos agora um algoritmo que tem como entradas as freqüências (que são as probabilidades de ocorrência) dos símbolos em uma seqüência e produz como saída um código de prefixo que codifica a seqüência usando o menor número possível de bits, entre todos os códigos de prefixo binários possíveis para estes símbolos. Este algoritmo, conhecido como **codificação de Huffman**, foi desenvolvido por David Huffman em um trabalho de conclusão de curso que escreveu em 1951, quando era estudante de pós-graduação no MIT. (Observe que este algoritmo supõe que já sabemos quantas vezes cada símbolo ocorre na seqüência, de modo que podemos calcular a freqüência de cada símbolo dividindo o número de vezes que ele ocorre pelo comprimento da seqüência.) A codificação de Huffman é um algoritmo fundamental na *compressão de dados*, o assunto dedicado a reduzir o número de bits necessários para representar informação. A codificação de Huffman é amplamente usada para comprimir seqüências de bits que representam textos e também desempenha um papel importante na compressão de arquivos de áudio e de imagens.

O Algoritmo 2 apresenta o algoritmo da codificação de Huffman. Dados os símbolos e suas freqüências, nosso objetivo é construir uma árvore binária enraizada, em que os símbolos são os rótulos das folhas. O algoritmo começa com uma floresta de árvores, cada uma das quais consis-

tindo em um vértice, em que cada vértice tem um símbolo como seu rótulo e em que o peso deste vértice é igual à freqüência do símbolo que é o seu rótulo. Em cada passo, combinamos duas árvores que tenham peso total mínimo em uma única árvore introduzindo uma nova raiz e colocando a árvore com maior peso total como sua subárvore esquerda e a árvore com menor peso como sua subárvore direita. Além disso, associamos a soma dos pesos das duas subárvores a esta árvore, como o peso total da árvore. (Embora procedimentos para desempates que escolhem entre árvores com igual peso possam ser especificados, não especificaremos esses procedimentos aqui.) O algoritmo termina quando for construída uma árvore, isto é, quando a floresta estiver reduzida a uma única árvore.

Demo

ALGORITMO 2 Codificação de Huffman.

procedure *Huffman*(C: símbolos a_i com freqüências w_i, $i = 1, ..., n$)
$F :=$ floresta de n árvores enraizadas, cada uma consistindo em um único vértice a_i com peso associado w_i
while F não é uma árvore
begin
  Substitua as árvores enraizadas T e T', de menores pesos de F, com $w(T) \geq w(T')$, por uma árvore que tenha uma nova raiz e que tenha T como subárvore esquerda e T'como subárvore direita. Rotule a nova aresta para T com 0 e a nova aresta para T' com 1.
  Associe $w(T) + w(T')$ como o peso da nova árvore
end
{a codificação de Huffman para os símbolos a_i é a concatenação dos rótulos das arestas no único caminho da raiz até o vértice a_i}

O Exemplo 5 ilustra como o Algoritmo 2 é usado para codificar um conjunto de seis símbolos.

EXEMPLO 5

Exemplos Extras

Use a codificação de Huffman para codificar os seguintes símbolos com as freqüências listadas: A: 0,08, B: 0,10, C: 0,12, D: 0,15, E: 0,20, F: 0,35. Qual é o número médio de bits usados para codificar um caractere?

Solução: A Figura 6 mostra os passos usados para codificar estes símbolos. O procedimento de codificação codifica A por 111, B por 110, C por 011, D por 010, E por 10 e F por 00. O número médio de bits usados para codificar um símbolo usando esta codificação é $3 \cdot 0{,}08 + 3 \cdot 0{,}10 + 3 \cdot 0{,}12 + 3 \cdot 0{,}15 + 2 \cdot 0{,}20 + 2 \cdot 0{,}35 = 2{,}45$. ◀

Links

DAVID A. HUFFMAN (1925–1999) David Huffman cresceu em Ohio. Com 18 anos de idade, formou-se em engenharia elétrica na Universidade Estadual de Ohio. Posteriormente, serviu na marinha americana como oficial de manutenção de radar em um destróier que teve a missão de limpar as minas nas águas asiáticas depois da Segunda Guerra Mundial. Mais tarde, obteve o seu mestrado na Universidade Estadual de Ohio e seu doutorado em engenharia elétrica no MIT. Huffman entrou para o corpo docente do MIT em 1953, onde permaneceu até 1967, quando se tornou membro fundador do departamento de Ciência da Computação na Universidade da Califórnia, em Santa Cruz. Ele desempenhou um papel importante no desenvolvimento deste departamento e permaneceu o restante de sua carreira lá, tendo se aposentado em 1994.

Huffman é conhecido por suas contribuições à teoria da informação e codificação, projetos de sinal para radar e para comunicações e desenvolvimento de procedimentos para circuitos lógicos assíncronos. Seu trabalho em superfícies com curvatura nula o levou a desenvolver técnicas originais para dobrar papel e vinil em formas não usuais, consideradas trabalho de arte por muitos e exibidas publicamente em diversas exposições. Entretanto, Huffman é bem mais conhecido pelo desenvolvimento do que agora é conhecido como Codificação de Huffman, resultado de um trabalho final de curso que ele escreveu durante sua pós-graduação no MIT.

Huffman apreciava exploração da natureza, caminhadas e viajar muito. Ele se tornou um mergulhador qualificado, quando tinha quase 70 anos. Ele mantinha serpentes venenosas como animais de estimação.

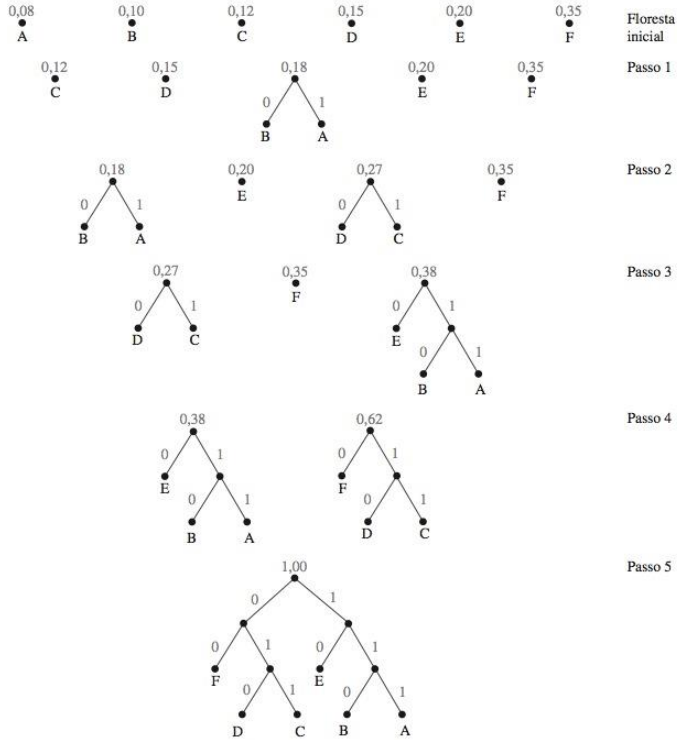


FIGURA 6 **Codificação de Huffman dos Símbolos no Exemplo 4.**

Observe que a codificação de Huffman é um algoritmo voraz. A substituição das duas subárvores com menor peso a cada passo leva a uma codificação ótima no sentido de que nenhum código de prefixo binário para estes símbolos pode codificá-los usando menos bits. Deixamos como Exercício 32, no final desta seção, a demonstração de que os códigos de Huffman são ótimos.

Existem muitas variações da codificação de Huffman. Por exemplo, em vez de codificar símbolos únicos, podemos codificar blocos de símbolos de um comprimento especificado, tais como blocos de dois símbolos. Fazer isso pode reduzir o número de bits necessários para codificar a seqüência (veja o Exercício 30 no final desta seção). Podemos também usar mais de dois símbolos para codificar os símbolos originais na seqüência (veja o preâmbulo do Exercício 28, no final desta seção). Além disso, uma variação conhecida como codificação de Huffman adaptativa (veja [Sa00]) pode ser usada quando a freqüência de cada símbolo em uma seqüência não é conhecida antecipadamente, de modo que a codificação é feita ao mesmo tempo em que a seqüência está sendo lida.

Árvores de Jogos

Links

As árvores têm sido usadas para analisar certos tipos de jogos, tais como jogo da velha, nim, damas e xadrez. Em cada um desses jogos, dois jogadores se alternam nos movimentos. Cada jogador conhece os movimentos feitos pelo outro jogador e nenhum elemento de sorte faz parte do jogo. Modelamos esses jogos usando **árvores de jogos**; os vértices dessas árvores representam as posições nas quais o jogo pode estar conforme progride; as arestas representam movimentos legais entre essas posições. Como as árvores de jogos em geral são grandes, simplificamo-nas pela representação de todas as posições simétricas de um jogo pelo mesmo vértice. Entretanto, a mesma posição de um jogo pode ser representada por vértices diferentes se seqüências diferentes de movimentos levarem a esta posição. A raiz representa a posição inicial. A convenção usual é representar vértices nos níveis pares por quadrados e vértices nos níveis ímpares por círculos. Quando o jogo está em uma posição representada por um vértice em um nível par, é a vez de o primeiro jogador jogar; quando o jogo está em uma posição representada por um nível ímpar, é a vez de o segundo jogador jogar. As árvores de jogos podem ser infinitas quando os jogos que elas representam nunca terminam, tais como jogos que entram em laços infinitos, mas a maioria dos jogos tem regras que levam a árvores de jogos finitas.

As folhas de uma árvore de jogo representam as posições finais de um jogo. Associamos um valor a cada folha, indicando o prêmio do primeiro jogador se o jogo terminar na posição representada por esta folha. Para jogos do tipo ganha ou perde, um vértice final, representado por um círculo, é rotulado 1 para indicar uma vitória do primeiro jogador, e um vértice final, representado por um quadrado, é rotulado -1 para indicar uma vitória do segundo jogador. Para jogos em que são permitidos empates, um vértice final, correspondente a uma posição de empate, é rotulado 0. Observe que para jogos ganha ou perde, associamos valores aos vértices finais, de modo que quanto maior o valor, melhor o resultado para o primeiro jogador.

No Exemplo 6, mostramos a árvore de jogo para um jogo bem conhecido e bem estudado.

EXEMPLO 6 **Nim** Em uma versão do jogo do **nim**, no começo de uma partida existem diversas pilhas de pedras. Dois jogadores se alternam nos movimentos; uma jogada legal consiste na remoção de uma ou mais pedras de uma das pilhas, sem remover todas as pedras restantes. O jogador que não faz um movimento legal perde. (Uma outra forma de ver isso é dizer que o jogador que remove a última pedra perde, pois a posição sem nenhuma pilha de pedras não é permitida.) A árvore de jogo mostrada na Figura 7 representa esta versão do nim, começando em uma posição inicial em

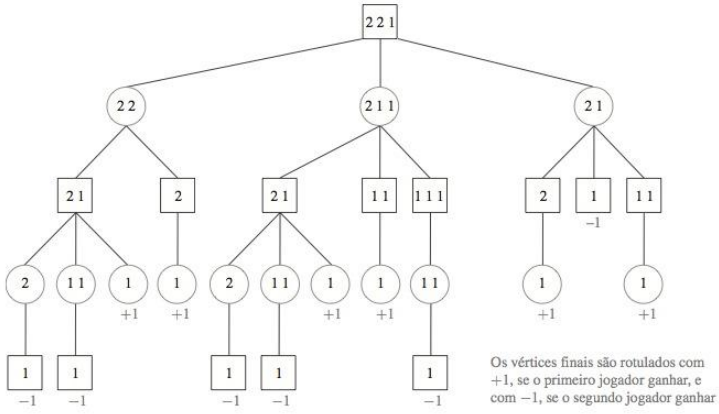


FIGURA 7 A Árvore de Jogo para um Jogo do Nim.

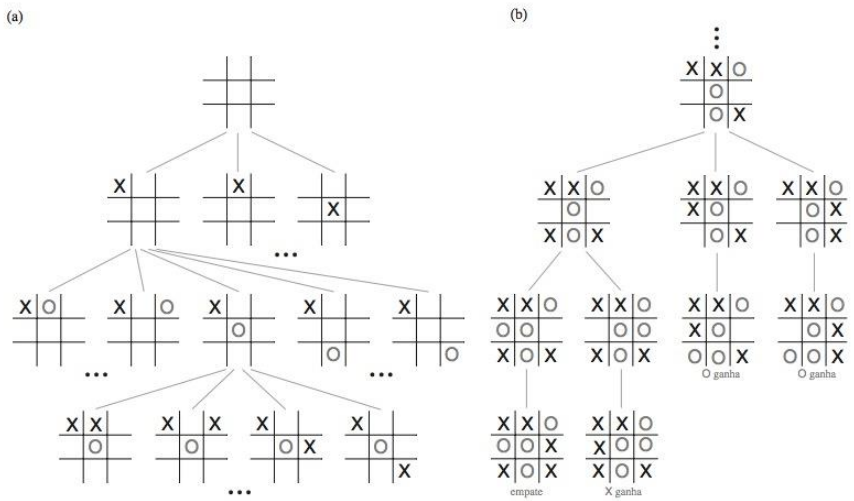


FIGURA 8 **Algumas das Árvores de Jogos para o Jogo da Velha.**

que há três pilhas de pedras com duas, duas e uma pedra cada, respectivamente. Representamos cada posição com uma lista não ordenada do número de pedras nas diferentes pilhas (a ordem das pilhas não importa). A jogada inicial do primeiro jogador pode levar a três posições possíveis, pois este jogador pode remover uma pedra de uma pilha com duas pedras (deixando três pilhas com uma, uma e duas pedras); duas pedras de uma pilha com duas pedras (deixando duas pilhas com duas e uma pedras); ou uma pedra da pilha com uma pedra (deixando duas pilhas de duas pedras). Quando ficar apenas uma pilha com uma pedra, nenhum movimento legal é possível, de modo que essa posição é final. Como o nim é um jogo ganha ou perde, rotulamos os vértices finais com $+1$ quando representarem vitória do primeiro jogador e com -1 quando representarem vitória do segundo jogador. ◀

EXEMPLO 7 **Jogo da velha** A árvore de jogo para o jogo da velha é extremamente grande e não pode ser desenhada aqui, embora um computador pudesse facilmente construir essa árvore. Mostramos parte do jogo da velha na Figura 8(a). Observe que, considerando posições simétricas como equivalentes, precisamos apenas considerar três possíveis movimentos iniciais, como mostra a Figura 8(a). Mostramos também uma subárvore desta árvore de jogo que leva às posições finais na Figura 8(b), em que um jogador que pode ganhar faz uma jogada vencedora. ◀

Podemos definir recursivamente o valor de todos os vértices em uma árvore de jogo de maneira que nos permita determinar o resultado deste jogo quando ambos os jogadores seguirem estratégias ótimas. Por **estratégia** queremos dizer um conjunto de regras que diz a um jogador como escolher jogadas para ganhar o jogo. Uma estratégia ótima para o primeiro jogador é uma estratégia que maximiza o prêmio deste jogador e para o segundo jogador é uma estratégia que minimiza este prêmio. Agora, definiremos recursivamente o valor de um vértice.

DEFINIÇÃO 1 O *valor de um vértice em uma árvore de jogo* é definido recursivamente como:
(*i*) o valor de uma folha é o prêmio do primeiro jogador quando o jogo termina na posição representada pela folha.
(*ii*) o valor de um vértice interno em um nível par é o máximo dos valores de seus filhos, e o valor de um vértice interno em um nível ímpar é o mínimo dos valores de seus filhos.

A estratégia na qual o primeiro jogador se move para uma posição representada por um filho com valor máximo e o segundo jogador se move para uma posição de um filho com valor mínimo é chamada de **estratégia minimax**. Podemos determinar quem vai ganhar o jogo quando ambos os jogadores seguirem a estratégia minimax pelo cálculo do valor da raiz da árvore; este valor é chamado de **valor** da árvore. Esta é uma conseqüência do Teorema 3.

TEOREMA 3 O valor de um vértice de uma árvore de jogo nos diz o prêmio do primeiro jogador se ambos os jogadores seguirem a estratégia minimax e o jogo começar da posição representada por este vértice.

Demonstração: Usaremos indução para demonstrar este teorema.

PASSO–BASE: Se o vértice for uma folha, por definição o valor associado a este vértice é o prêmio do primeiro jogador.

PASSO DE INDUÇÃO: A hipótese de indução é a suposição de que os valores dos filhos de um vértice são os prêmios do primeiro jogador, supondo que o jogo comece em cada uma das posições representadas por estes vértices. Precisamos considerar dois casos, quando é a vez do primeiro jogador e quando é a vez do segundo jogador.

Quando for a vez do primeiro jogador, este jogador segue a estratégia minimax e se move para a posição representada pelo filho com maior valor. Pela hipótese de indução, este valor é o prêmio do primeiro jogador quando o jogo começa na posição representada por este filho e segue a estratégia minimax. Pelo passo recursivo na definição do valor de um vértice interno em um nível par (como o valor máximo de seus filhos), o valor deste vértice é o prêmio quando o jogo começa na posição representada por este vértice.

Quando for a vez do segundo jogador, este jogador segue a estratégia minimax e se move para a posição representada pelo filho com menor valor. Pela hipótese de indução, este valor é o prêmio do primeiro jogador quando o jogo começa na posição representada por este filho e ambos os jogadores seguem a estratégia minimax. Pelo passo recursivo na definição do valor de um vértice interno em um nível ímpar como o valor mínimo de seus filhos, o valor deste vértice é o prêmio quando o jogo começa na posição representada por este vértice. ◁

Lembre-se: Estendendo a demonstração do Teorema 3, pode ser mostrado que a estratégia minimax é a estratégia ótima para ambos os jogadores.

O Exemplo 8 ilustra como o procedimento minimax funciona. Ele mostra os valores associados aos vértices internos da árvore de jogo do Exemplo 6. Note que podemos diminuir os cálculos necessários, observando que para jogos ganha ou perde, uma vez que um filho de um vértice quadrado com valor $+1$ seja encontrado, o valor do vértice quadrado também é $+1$, pois $+1$ é o maior prêmio possível.

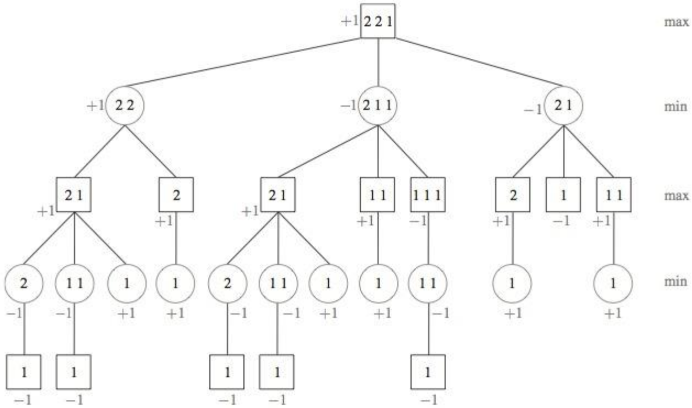


FIGURA 9 Mostrando os Valores dos Vértices no Jogo do Nim.

De maneira análoga, uma vez que um filho de um vértice círculo com valor -1 seja encontrado, este também é o valor do vértice círculo.

EXEMPLO 8 No Exemplo 6, construímos a árvore de jogo para o nim com uma posição de partida em que existem três pilhas com duas, duas e uma pedra. Na Figura 9, mostramos os valores dos vértices desta árvore de jogo. Os valores dos vértices são calculados usando os valores das folhas e trabalhando um nível para cima de cada vez. Na margem direita desta figura, indicamos se usamos o máximo ou o mínimo dos valores dos filhos para encontrar o valor de um vértice interno em cada nível. Por exemplo, uma vez que tivermos encontrado os valores dos três filhos da raiz, que são $1, -1$ e -1, encontramos o valor da raiz calculando $\max(1, -1, -1) = 1$. Como o valor da raiz é 1, segue que o primeiro jogador ganha quando ambos os jogadores seguirem a estratégia minimax. ◀

Árvores de jogos para alguns jogos bem conhecidos podem ser extraordinariamente grandes, pois estes jogos têm muitas jogadas possíveis diferentes. Por exemplo, foi feita uma estimativa de que a árvore de jogo para o xadrez tem 10^{100} vértices! Pode ser impossível usar o Teorema 3 diretamente para estudar um jogo por causa do tamanho da árvore de jogo. Assim, várias abordagens foram desenvolvidas para ajudar a determinar boas estratégias e para determinar o resultado desses jogos. Uma técnica útil, chamada de *poda alpha-beta* (*alpha-beta pruning*, em inglês), elimina muitos cálculos, cortando partes da árvore de jogo que não podem afetar os valores dos vértices ancestrais. (Para informação sobre a poda alpha-beta, consulte [Gr90].) Uma outra abordagem útil é usar *funções de avaliação*, que estimam o valor dos vértices internos em uma árvore de jogo quando não for viável calcular esses valores exatamente. Por exemplo, no jogo da velha, como uma função de avaliação para uma posição, podemos usar o número de filas (linhas, colunas e diagonais) que não contêm nenhum O (usado para indicar jogadas do segundo jogador) menos o número de filas que não contêm nenhum X (usado para indicar jogadas do primeiro jogador). Esta função de avaliação fornece alguma indicação de qual jogador tem a vantagem no jogo. Uma vez que os valores de uma função de avaliação são inseridos, o valor do jogo pode ser calculado seguindo as regras usadas na estratégia minimax. Programas computacionais criados para jogar xadrez, tal como o famoso programa Deep Blue, baseiam-se em funções de avaliação sofisticadas. Para mais informação sobre como os computadores jogam xadrez, veja [Le91].

Links

Exercícios

1. Construa uma árvore de busca binária para as palavras *banana, peach, apple, pear, coconut, mango* e *papaya* usando a ordem alfabética.
2. Construa uma árvore de busca binária para as palavras *oenology phrenology, campanology, ornithology, ichthyology, limnology, alchemy* e *astrology* usando a ordem alfabética.
3. Quantas comparações são necessárias para localizar ou adicionar cada uma das palavras na árvore de busca do Exercício 1, começando novamente em cada vez?
   a) *pear* **b)** *banana*
   c) *kumquat* **d)** *orange*
4. Quantas comparações são necessárias para localizar ou adicionar cada uma das palavras na árvore de busca do Exercício 2, começando novamente em cada vez?
   a) *palmistry* **b)** *etymology*
   c) *paleontology* **d)** *glaciology*
5. Usando a ordem alfabética, construa uma árvore de busca binária para as palavras na sentença "*The quick brown fox jumps over the lazy dog*."
6. Quantas pesagens em uma balança de pratos são necessárias para encontrar uma moeda falsa mais leve entre quatro moedas? Descreva um algoritmo para encontrar a moeda mais leve usando este número de pesagens.
7. Quantas pesagens em uma balança de pratos são necessárias para encontrar uma moeda falsa entre quatro moedas se a moeda falsa puder tanto ser mais pesada quanto mais leve do que as outras? Descreva um algoritmo para encontrar a moeda falsa usando este número de pesagens.

* **8.** Quantas pesagens em uma balança de pratos são necessárias para encontrar uma moeda falsa entre oito moedas se a moeda falsa puder tanto ser mais pesada quanto mais leve do que as outras? Descreva um algoritmo para encontrar a moeda falsa usando este número de pesagens.

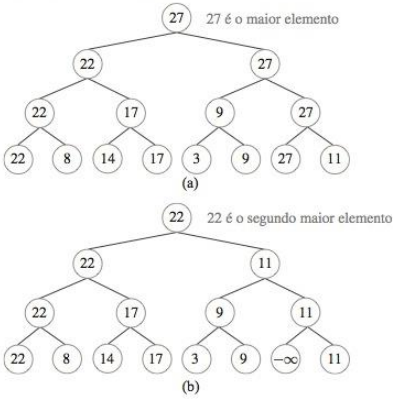
***9.** Quantas pesagens em uma balança de pratos são necessárias para encontrar uma moeda falsa entre 12 moedas se a moeda falsa for mais leve do que as outras? Descreva um algoritmo para encontrar a moeda mais leve usando este número de pesagens.

***10.** Uma de quatro moedas pode ser falsa. Se ela for falsa, pode ser mais leve ou mais pesada do que as outras. Quantas pesagens são necessárias, usando uma balança de pratos, para determinar se existe uma moeda falsa e, se existir, se ela é mais leve ou mais pesada do que as outras? Descreva um algoritmo para encontrar a moeda falsa e determinar se ela é mais leve ou mais pesada do que as outras usando este número de pesagens.

11. Encontre o número mínimo de comparações necessárias para ordenar quatro elementos e desenvolva um algoritmo que ordene esses elementos usando este número de comparações.

***12.** Encontre o número mínimo de comparações necessárias para ordenar cinco elementos e desenvolva um algoritmo que ordene esses elementos usando este número de comparações.

A **ordem de torneio** é um algoritmo de ordenação que funciona pela construção de uma árvore binária ordenada. Representamos os elementos a serem ordenados pelos vértices que se tornarão as folhas. Construímos a árvore um nível de cada vez como construiríamos a árvore que representa os vencedores em um torneio. Trabalhando da esquerda para a direita, comparamos pares de elementos consecutivos, adicionando um vértice pai rotulado com o maior dos dois elementos em comparação. Fazemos comparações análogas entre os rótulos dos vértices em cada nível até que atinjamos a raiz da árvore que é rotulada com o maior elemento. A árvore construída pela ordem de torneio de 22, 8, 14, 17, 3, 9, 27, 11 é ilustrada na parte (a) da figura. Uma vez que o maior elemento tenha sido determinado, a folha com este rótulo é rotulada novamente por $-\infty$, que é definido como menor do que qualquer elemento. Os rótulos de todos os vértices no caminho deste vértice até a raiz da árvore são recalculados, como mostra a parte (b) da figura. Isso produz o segundo maior elemento. Este processo continua até que toda a lista tenha sido ordenada.



13. Complete a ordem de torneio da lista 22, 8, 14, 17, 3, 9, 27, 11. Mostre os rótulos dos vértices em cada passo.

14. Use a ordem de torneio para ordenar a lista 17, 4, 1, 5, 13, 10, 14, 6.

15. Descreva a ordem de torneio usando pseudocódigo.

16. Supondo que n, o número de elementos a serem ordenados, seja igual a 2^k para algum inteiro positivo k, determine o número de comparações usadas na ordem de torneio para encontrar o maior elemento da lista usando a ordem de torneio.

17. Quantas comparações a ordem de torneio usa para encontrar o segundo maior elemento, o terceiro maior elemento e assim por diante, até o $(n-1)$-ésimo maior (ou o segundo menor) elemento?

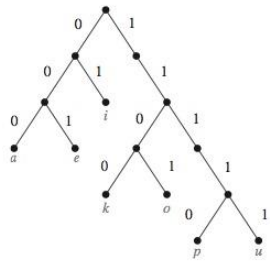
18. Mostre que a ordem de torneio exige $\Theta(n \log n)$ comparações para ordenar uma lista de n elementos. [*Dica:* Pela inserção de um número adequado de elementos fictícios definido

como menor do que todos os inteiros, tal como $-\infty$, suponha que $n = 2^k$ para algum inteiro positivo k.]

19. Quais destes códigos são códigos de prefixo?
a) a: 11, e: 00, t: 10, s: 01
b) a: 0, e: 1, t: 01, s: 001
c) a: 101, e: 11, t: 001, s: 011, n: 010
d) a: 010, e: 11, t: 011, s: 1011, n: 1001, i: 10101

20. Construa a árvore binária com os códigos de prefixo que representam estes esquemas de codificação.
a) a: 11, e: 0, t: 101, s: 100
b) a: 1, e: 01, t: 001, s: 0001, n: 00001
c) a: 1010, e: 0, t: 11, s: 1011, n: 1001, i: 10001

21. Quais são os códigos para a, e, i, k, o, p e u se o esquema de codificação for representado por esta árvore?



22. Dado o esquema de codificação a: 001, b: 0001, e: 1, r: 0000, s: 0100, t: 011, x: 01010, encontre a palavra representada por
a) 01110100011. **b)** 0001110000.
c) 0100101010. **c)** 01100101010.

23. Use a codificação de Huffman para codificar estes símbolos com as freqüências dadas: a: 0,20, b: 0,10, c: 0,15, d: 0,25, e: 0,30. Qual o número médio de bits necessários para codificar um caractere?

24. Use a codificação de Huffman para codificar estes símbolos com as freqüências dadas: A: 0,10, B: 0,25, C: 0,05, D: 0,15, E: 0,30, F: 0,07, G: 0,08. Qual o número médio de bits necessários para codificar um caractere?

25. Construa dois códigos de Huffman diferentes para estes símbolos e freqüências: t: 0,2, u: 0,3, v: 0,2, w: 0,3.

26. **a)** Use a codificação de Huffman para codificar estes símbolos com freqüências: a: 0,4, b: 0,2, c: 0,2, d: 0,1, e: 0,1 de duas maneiras diferentes, obtidas desempatando situações de empate de modo diferente no algoritmo. Primeiro, entre todas as árvores de peso mínimo, escolha duas árvores com o maior número de vértices para combinar em cada estágio do algoritmo. Segundo, entre as árvores de peso mínimo, escolha duas árvores com o menor número de vértices em cada estágio.
b) Calcule o número médio de bits necessários para codificar um símbolo com cada código e calcule a variância deste número de bits para cada código. Que procedimento de desempate produz a menor variância no número de bits necessários para codificar um símbolo?

27. Construa um código de Huffman para as letras do alfabeto inglês em que as freqüências das letras em um texto típico em inglês são mostradas na tabela.

Letra	*Freqüência*	*Letra*	*Freqüência*
A	0,0817	N	0,0662
B	0,0145	O	0,0781
C	0,0248	P	0,0156
D	0,0431	Q	0,0009
E	0,1232	R	0,0572
F	0,0209	S	0,0628
G	0,0182	T	0,0905
H	0,0668	U	0,0304
I	0,0689	V	0,0102
J	0,0010	W	0,0264
K	0,0080	X	0,0015
L	0,0397	Y	0,0211
M	0,0277	Z	0,0005

Suponha que m seja um inteiro positivo com $m \geq 2$. Um código de Huffman de grau m para um conjunto de N símbolos pode ser construído de modo análogo à construção de um código de Huffman binário. No passo inicial, $((N-1) \bmod (m-1)) + 1$ árvores que consistem em um único vértice com pesos menores são combinadas em uma árvore enraizada com estes vértices como folhas. Em cada passo subseqüente, as m árvores de peso menor são combinadas em uma árvore de grau m.

28. Descreva um algoritmo para a codificação de Huffman de grau m em pseudocódigo.

29. Usando os símbolos 0, 1 e 2, use um código de Huffman ternário ($m = 3$) para codificar estas letras com as freqüências dadas: A: 0,25, E: 0,30, N: 0,10, R: 0,05, T: 0,12, Z: 0,18.

30. Considere os três símbolos A, B e C com freqüências A: 0,80, B: 0,19, C: 0,01.
a) Construa um código de Huffman para estes três símbolos.
b) Forme um novo conjunto de nove símbolos agrupando blocos de dois símbolos, AA, AB, AC, BA, BB, BC, CA, CB e CC. Construa um código de Huffman para estes nove símbolos, supondo que a ocorrência de símbolos no texto original seja independente.
c) Compare o número médio de bits necessários para codificar texto usando o código de Huffman para três símbolos da parte (a) e o código de Huffman para nove blocos de dois símbolos construído na parte (b). Qual é mais eficiente?

31. Dados $n+1$ símbolos $x_1, x_2, \ldots, x_n, x_{n+1}$ que aparecem 1, $f_1, f_2, \ldots, f_n$ vezes em uma seqüência de símbolos, respectivamente, em que f_j é o j-ésimo número de Fibonacci, qual é o número máximo de bits usados para codificar um símbolo quando todas as escolhas de desempate possíveis são consideradas em cada estágio do algoritmo de codificação de Huffman?

*32. Mostre que os códigos de Huffman são ótimos no sentido de que representam uma seqüência de símbolos que usa o mínimo de bits entre todos os códigos de prefixo binários.

33. Desenhe uma árvore de jogo para o nim se a posição inicial consiste em duas pilhas de duas e três pedras, respectivamente. Quando desenhar a árvore, represente pelo mesmo vértice posições simétricas que resultem de uma mesma jogada. Encontre os valores de cada vértice da árvore de jogo. Quem ganha o jogo se ambos os jogadores seguirem uma estratégia ótima?

34. Desenhe uma árvore de jogo para o nim se a posição inicial consiste em três pilhas de uma, duas e três pedras, respectivamente. Quando desenhar a árvore, represente pelo mesmo vértice posições simétricas que resultem de uma mesma jogada. Encontre os valores de cada vértice da árvore de jogo. Quem ganha o jogo se ambos os jogadores seguirem uma estratégia ótima?

35. Suponha que variemos o prêmio do jogador vencedor no jogo do Nim, de modo que o prêmio seja n dólares quando n for o número de jogadas legais feitas antes de uma posição final ser atingida. Encontre o prêmio do primeiro jogador se a posição inicial consistir em
   a) duas pilhas com uma e três pedras, respectivamente.
   b) duas pilhas com duas e quatro pedras, respectivamente.
   c) três pilhas com uma, duas e três pedras, respectivamente.

36. Suponha que em uma variação do jogo do nim, permitamos que um jogador remova uma ou mais pedras de uma pilha ou que junte as pedras de duas pilhas em uma única pilha contanto que permaneça pelo menos uma pedra. Desenhe a árvore de jogo para esta variação do nim se a posição inicial consistir em três pilhas com duas, duas e uma pedra, respectivamente. Encontre os valores de cada vértice na árvore de jogo e determine o vencedor se ambos os jogadores seguirem uma estratégia ótima.

37. Desenhe a subárvore para o jogo da velha, começando em cada uma destas posições. Determine o valor de cada uma destas subárvores.

a)

O	X	X
X	O	O
		X

b)

X	O	X
O	X	X
		O

c)

X		O
O	O	
X		X

d)

	O	X
	X	O
X	O	

38. Suponha que as primeiras quatro jogadas de um jogo da velha sejam como as mostradas. O primeiro jogador (cujas jogadas são marcadas por X) tem uma estratégia que sempre vencerá?

a)

X	O	X
	O	

b)

X	O	X
	O	

c)

	O	X
	X	
O		

d)

X		
X	O	
O		

39. Mostre que se um jogo do nim começar com duas pilhas com o mesmo número de pedras, contanto que este número seja pelo menos dois, então o segundo jogador ganha quando ambos os jogadores seguirem estratégias ótimas.

40. Mostre que se um jogo do nim começar com duas pilhas com números diferentes de pedras, o primeiro jogador ganha quando ambos os jogadores seguirem estratégias ótimas.

41. Quantos filhos tem a raiz da árvore de jogo do jogo de damas? Quantos netos ela tem?

42. Quantos filhos tem a raiz da árvore de jogo do nim e quantos netos ela tem, se a posição inicial for
   a) pilhas com quatro e cinco pedras, respectivamente.
   b) pilhas com duas, três e quatro pedras, respectivamente.
   c) pilhas com uma, duas, três e quatro pedras, respectivamente.
   d) pilhas com duas, duas, três, três e cinco pedras, respectivamente.

43. Desenhe a árvore de jogo para o jogo da velha para os níveis correspondentes às duas primeiras jogadas. Determine o valor da função de avaliação mencionada no texto que associa a uma posição o número de filas que não contém nenhum 0 menos o número de filas que não contém nenhum X como o valor de cada vértice naquele nível e calcule o valor desta árvore para os vértices como se a função de avaliação desse os valores corretos destes vértices.

44. Use pseudocódigo para descrever um algoritmo que determine o valor de uma árvore de jogo quando ambos os jogadores seguirem uma estratégia minimax.

10.3 Percurso em Árvores

Introdução

Árvores enraizadas ordenadas são freqüentemente usadas para armazenar informações. Precisamos de procedimentos para visitar cada vértice de uma árvore enraizada ordenada para acessar dados. Descreveremos diversos algoritmos importantes para visitar todos os vértices de uma árvore enraizada ordenada. Árvores enraizadas ordenadas também podem ser usadas para represen-

tar diversos tipos de expressões, tais como expressões aritméticas que envolvem números, variáveis e operações. As diferentes listagens dos vértices de uma árvore enraizada ordenada usadas para representar expressões são úteis no cálculo dessas expressões.

Sistemas de Endereçamento Globais

Procedimentos para percorrer todos os vértices de uma árvore enraizada ordenada baseiam-se na ordem dos filhos. Nas árvores enraizadas ordenadas, os filhos de um vértice interno são mostrados da esquerda para a direita nos desenhos que representam estes grafos orientados.

Descreveremos uma maneira de ordenar totalmente os vértices de uma árvore enraizada ordenada. Para produzir esta ordem, devemos inicialmente rotular todos os vértices. Fazemos isso recursivamente:

1. Rotule a raiz com o inteiro zero. A seguir, rotule seus k filhos (no nível 1) da esquerda para a direita, com 1, 2, 3, …, k.
2. Para cada vértice v no nível n com rótulo A, rotule seus k_v filhos conforme eles forem desenhados da esquerda para a direita, com $A.1, A.2, \ldots, A.k_v$.

Seguindo este procedimento, um vértice v no nível n, para $n \geq 1$, é rotulado $x_1 \cdot x_2 \ldots x_n$, em que o único caminho da raiz a v passa pelo x_1-ésimo vértice no nível 1, pelo x_2-ésimo vértice no nível 2, e assim por diante. Esta rotulação é chamada de **sistema de endereçamento global** da árvore enraizada ordenada.

Podemos ordenar totalmente os vértices usando a ordem lexicográfica de seus rótulos no sistema de endereçamento global. O vértice rotulado $x_1 \cdot x_2 \ldots . x_n$ é menor do que o vértice rotulado $y_1 \cdot y_2 \ldots y_m$ se existir um i, $0 \leq i \leq n$, com $x_1 = y_1, x_2 = y_2, \ldots, x_{i-1} = y_{i-1}$ e $x_i < y_i$; ou se $n < m$ e $x_i = y_i$ para $i = 1, 2, \ldots, n$.

EXEMPLO 1 Mostramos a rotulação do sistema de endereçamento global próximo aos vértices na árvore enraizada ordenada ilustrada na Figura 1. A ordem lexicográfica da rotulação é

Exemplos Extras

$$0 < 1 < 1.1 < 1.2 < 1.3 < 2 < 3 < 3.1 < 3.1.1 < 3.1.2 < 3.1.2.1 < 3.1.2.2 < 3.1.2.3 < 3.1.2.4 < 3.1.3 < 3.2 < 4 < 4.1 < 5 < 5.1 < 5.1.1 < 5.2 < 5.3$$ ◀

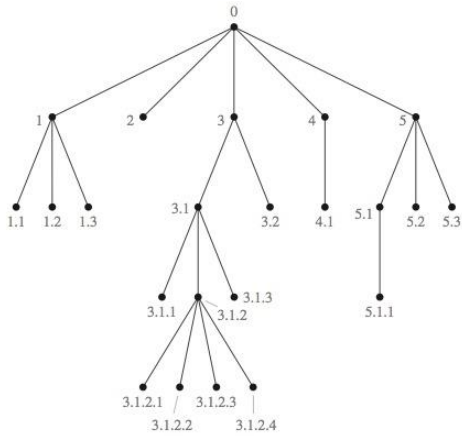


FIGURA 1 O Sistema de Endereçamento Global de uma Árvore Enraizada Ordenada.

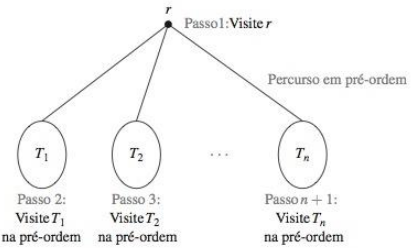


FIGURA 2 Percurso em Pré-ordem.

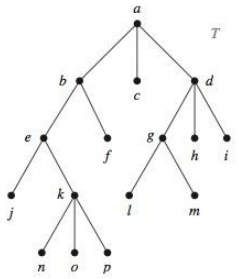


FIGURA 3 A Árvore Enraizada Ordenada *T*.

Algoritmos de Percurso

Procedimentos para visitar sistematicamente todos os vértices de uma árvore enraizada ordenada são chamados de **algoritmos de percurso**. Descreveremos três dos mais comumente usados desses algoritmos, **percurso em pré-ordem**, **percurso em ordem simétrica** e **percurso em pós-ordem**. Cada um destes algoritmos pode ser definido recursivamente. Definimos primeiro o percurso em pré-ordem.

DEFINIÇÃO 1 Seja T uma árvore enraizada ordenada com raiz r. Se T consistir apenas em r, então r é o *percurso em pré-ordem* de T. Caso contrário, suponha que $T_1, T_2, \ldots, T_n$ sejam subárvores em r da esquerda para a direita em T. O *percurso em pré-ordem* começa visitando r. Ele continua percorrendo T_1 em pré-ordem, a seguir T_2 em pré-ordem e assim por diante, até que T_n seja percorrida em pré-ordem.

O leitor deve verificar que o percurso em pré-ordem de uma árvore enraizada ordenada dá a mesma ordem dos vértices que a ordem obtida usando um sistema de endereçamento global. A Figura 2 indica como o percurso em pré-ordem é feito.

O Exemplo 2 ilustra o percurso em pré-ordem.

EXEMPLO 2 Em qual ordem um percurso em pré-ordem visita os vértices na árvore enraizada ordenada T mostrada na Figura 3?

Exemplos Extras

Solução: Os passos do percurso em pré-ordem de T são mostrados na Figura 4. Percorremos T em pré-ordem listando primeiro a raiz a, seguida pela lista em pré-ordem da subárvore com raiz b, a lista em pré-ordem da subárvore com raiz c (que é apenas c) e a lista em pré-ordem da subárvore com raiz d.

A lista em pré-ordem da subárvore com raiz b começa listando b, a seguir os vértices da subárvore com raiz e em pré-ordem e a seguir a subárvore com raiz f em pré-ordem (que é apenas f). A lista em pré-ordem da subárvore com raiz d começa listando d, seguida da lista em pré-ordem da subárvore com raiz g, seguida da subárvore com raiz h (que é apenas h), seguida da subárvore com raiz i (que é apenas i).

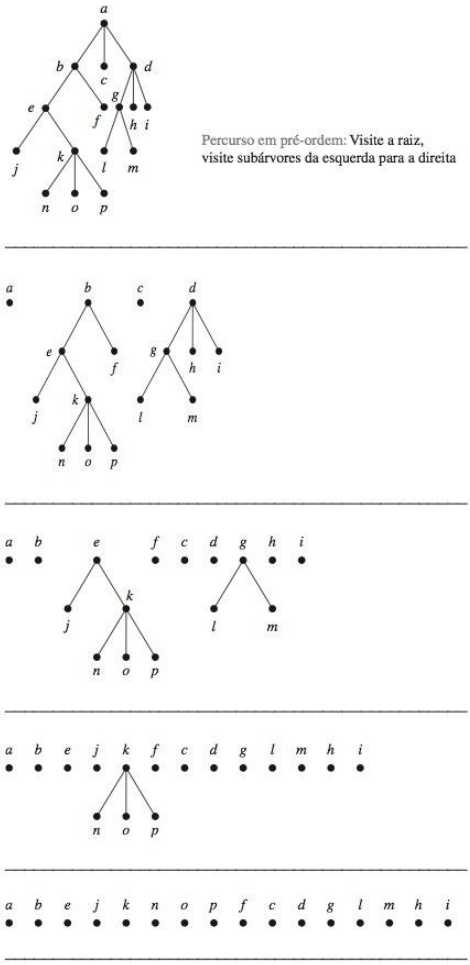


FIGURA 4 O Percurso em Pré-ordem de *T*.

A lista em pré-ordem da subárvore com raiz *e* começa listando *e*, seguida pela listagem em pré-ordem da subárvore com raiz *j* (que é apenas *j*), seguida pela listagem em pré-ordem da subárvore com raiz *k*. A listagem em pré-ordem da subárvore com raiz *g* é *g*, seguida por *l*, seguida por *m*. A listagem em pré-ordem da subárvore com raiz *k* é *k*, *n*, *o*, *p*. Conseqüentemente, o percurso em pré-ordem de *T* é *a*, *b*, *e*, *j*, *k*, *n*, *o*, *p*, *f*, *c*, *d*, *g*, *l*, *m*, *h*, *i*. ◀

Definiremos agora o percurso em ordem simétrica.

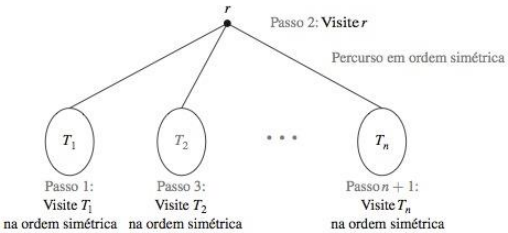


FIGURA 5 **Percurso em Ordem Simétrica.**

DEFINIÇÃO 2 Seja T uma árvore enraizada ordenada com raiz r. Se T consistir apenas em r, então r é o *percurso em ordem simétrica* de T. Caso contrário, suponha que $T_1, T_2, \ldots, T_n$ sejam subárvores em r da esquerda para a direita. O *percurso em ordem simétrica* começa percorrendo T_1 em ordem simétrica, visitando a seguir r. Ele continua percorrendo T_2 em ordem simétrica, a seguir, T_3 em ordem simétrica, ..., e finalmente T_n em ordem simétrica.

A Figura 5 indica como o percurso em ordem simétrica é feito. O Exemplo 3 ilustra como o percurso em ordem simétrica é feito para uma árvore particular.

EXEMPLO 3 Em qual ordem um percurso em ordem simétrica visita os vértices de uma árvore enraizada ordenada T na Figura 3?

Exemplos Extras

Solução: Os passos do percurso em ordem simétrica de uma árvore enraizada ordenada T são mostrados na Figura 6. O percurso em ordem simétrica começa com um percurso em ordem simétrica da subárvore com raiz b, a raiz a, a listagem em ordem simétrica da subárvore com raiz c, que é apenas c, e a listagem em ordem simétrica da subárvore com raiz d.

A listagem em ordem simétrica da subárvore com raiz b começa com a listagem em ordem simétrica da subárvore com raiz e, a raiz b, e f. A listagem em ordem simétrica da subárvore com raiz d começa com a listagem em ordem simétrica da subárvore com raiz g, seguida pela raiz d, seguida por h, seguida por i.

A listagem em ordem simétrica da subárvore com raiz e é j, seguida pela raiz e, seguida pela listagem em ordem simétrica da subárvore com raiz k. A listagem em ordem da subárvore com raiz g é l, g, m. A listagem em ordem simétrica da subárvore com raiz k é n, k, o, p. Conseqüentemente, a listagem em ordem simétrica da árvore enraizada ordenada é $j, e, n, k, o, p, b, f, a, c, l, g, m, d, h, i$. ◀

Definimos agora o percurso em pós-ordem.

DEFINIÇÃO 3 Seja T uma árvore enraizada ordenada com raiz r. Se T consistir apenas em r, então r é o *percurso em pós-ordem de T*. Caso contrário, suponha que $T_1, T_2, \ldots, T_n$ são as subárvores em r da esquerda para a direita. O *percurso em pós-ordem* começa percorrendo T_1 em pós-ordem, a seguir, T_2 em pós-ordem, ..., a seguir, T_n em pós-ordem, e termina visitando r.

A Figura 7 ilustra como o percurso em pós-ordem é feito. O Exemplo 4 ilustra como o percurso em pós-ordem funciona.

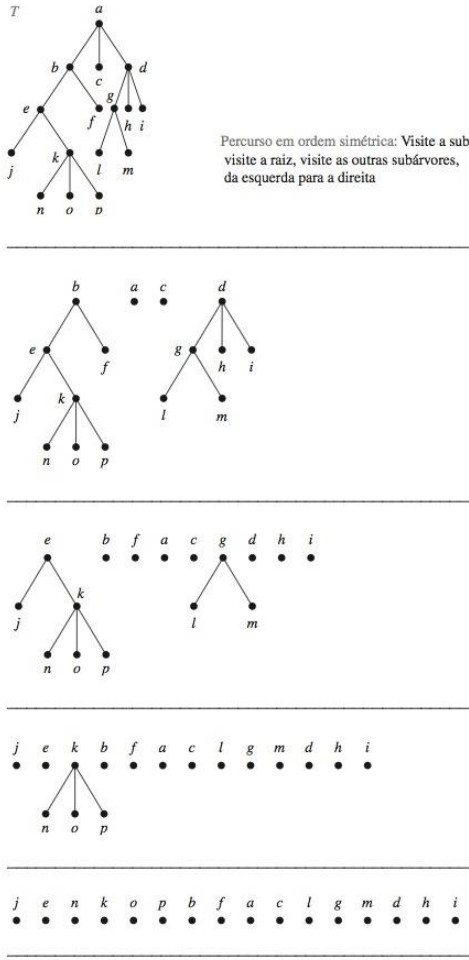


FIGURA 6 **O Percurso em Ordem Simétrica de *T*.**

EXEMPLO 4 Em qual ordem um percurso em pós-ordem visita os vértices da árvore enraizada ordenada T mostrada na Figura 3?

Exemplos Extras

Solução: Os passos do percurso em pós-ordem da árvore enraizada ordenada T são mostrados na Figura 8. O percurso em pós-ordem começa com o percurso em pós-ordem da subárvore com raiz

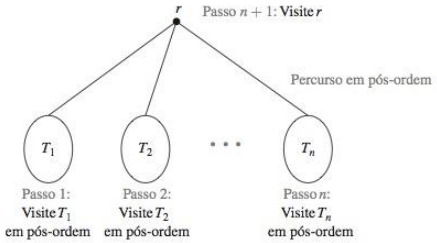


FIGURA 7 Percurso em Pós-ordem.

b, o percurso em pós-ordem da subárvore com raiz *c*, que é apenas *c*, o percurso em pós-ordem da subárvore com raiz *d*, seguido pela raiz *a*.

O percurso em pós-ordem da subárvore com raiz *b* começa com o percurso em pós-ordem da subárvore com raiz *e*, seguido por *f*, seguido pela raiz *b*. O percurso em pós-ordem da árvore enraizada com raiz *d* começa com o percurso em pós-ordem da subárvore com raiz *g*, seguida por *h*, seguida por *i*, seguida pela raiz *d*.

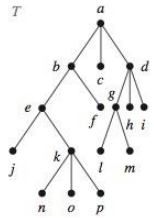
O percurso em pós-ordem da subárvore com raiz *e* começa com *j*, seguido pelo percurso em pós-ordem da subárvore com raiz *k*, seguido pela raiz *e*. O percurso em pós-ordem da subárvore com raiz *g* é *l*, *m*, *g*. O percurso em pós-ordem da subárvore com raiz *k* é *n*, *o*, *p*, *k*. Portanto, o percurso em pós-ordem de *T* é *j*, *n*, *o*, *p*, *k*, *e*, *f*, *b*, *c*, *l*, *m*, *g*, *h*, *i*, *d*, *a*. ◀

Existem maneiras fáceis de listar os vértices de uma árvore enraizada ordenada em pré-ordem, em ordem simétrica e em pós-ordem. Para fazer isso, primeiro desenhe uma curva em torno da árvore enraizada ordenada começando na raiz, movendo-se ao longo das arestas, como mostrado no exemplo na Figura 9. Podemos listar os vértices em pré-ordem listando cada vértice na primeira vez que a curva passa por ele. Podemos listar os vértices na ordem simétrica listando uma folha na primeira vez que a curva passa por ela e listando cada vértice interno na segunda vez que a curva passa por ele. Podemos listar os vértices em pós-ordem listando um vértice na última vez que ele é visitado no caminho de volta para seu pai. Quando isto é feito na árvore enraizada na Figura 9, segue que o percurso em pré-ordem é *a*, *b*, *d*, *h*, *e*, *i*, *j*, *c*, *f*, *g*, *k*, o percurso em ordem simétrica é *h*, *d*, *b*, *i*, *e*, *j*, *a*, *f*, *c*, *k*, *g*, e o percurso em pós-ordem é *h*, *d*, *i*, *j*, *e*, *b*, *f*, *k*, *g*, *c*, *a*.

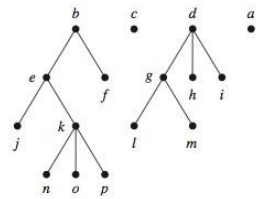
Algoritmos para percorrer árvores enraizadas ordenadas em pré-ordem, ordem simétrica e pós-ordem são mais facilmente expressos de modo recursivo.

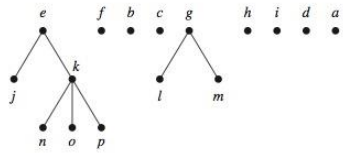
ALGORITMO 1 Percurso em Pré-ordem.

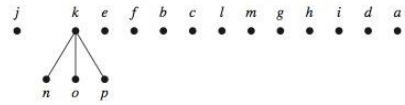
```
procedure pre-ordem(T: árvore enraizada ordenada)
r := raiz de T
liste r
for cada filho c de r da esquerda para a direita
begin
   T(c):= subárvore com c como sua raiz
   pre-ordem(T(c))
end
```



Percurso em pós-ordem: Visite subárvores da esquerda para a direita; visite a raiz







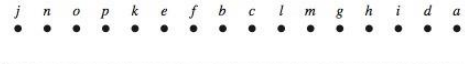


FIGURA 8 O Percurso em Pós-ordem de *T*.

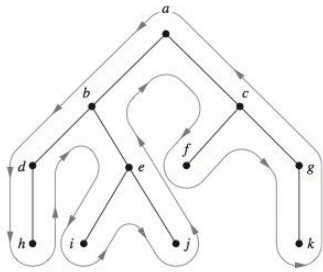


FIGURA 9 Um Atalho para Percorrer uma Árvore Enraizada Ordenada em Pré-ordem, Ordem Simétrica e Pós-ordem.

ALGORITMO 2 Percurso em Ordem Simétrica.

```
procedure ordem simetrica(T: árvore enraizada ordenada)
r := raiz de T
if r for uma folha then liste r
else
begin
  l := primeiro filho de r da esquerda para a direita
  T (l) := subárvore com l como sua raiz
  ordem simétrica(T(l))
  liste r
  for cada filho c de r, exceto l, da esquerda para a direita
    T (c) := subárvore com c como sua raiz
    ordem simetrica(T(c))
end
```

ALGORITMO 3 Percurso em Pós-ordem.

```
procedure pos-ordem(T: árvore enraizada ordenada)
r := raiz de T
for cada filho c de r da esquerda para a direita
begin
  T (c) := subárvore com c como sua raiz
  pos-ordem(T(c))
end
liste r
```

Observe que tanto o percurso em pré-ordem quanto o percurso em pós-ordem codificam a estrutura de uma árvore enraizada ordenada quando o número de filhos de cada vértice for especificado. Isto é, uma árvore enraizada ordenada é determinada de modo único quando especificamos uma lista de vértices gerada por um percurso em pré-ordem ou por um percurso em pós-ordem da árvore, junto com o número de filhos de cada vértice (veja os exercícios 26 e 27). Em particular,

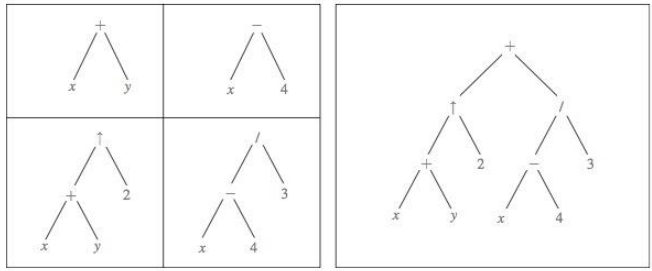


FIGURA 10 **Uma Árvore Binária que Representa $((x + y) \uparrow 2) + ((x - 4)/3)$.**

tanto o percurso em pré-ordem quanto o percurso em pós-ordem codificam a estrutura de uma árvore de grau m ordenada cheia. Entretanto, quando o número de filhos dos vértices não é especificado, nem um percurso em pré-ordem nem um percurso em pós-ordem codificam a estrutura de uma árvore enraizada ordenada (veja os exercícios 28 e 29).

Notação Infixa, Prefixa e Pós-fixa

Podemos representar expressões complicadas, tal como proposições compostas, combinações de conjuntos e expressões aritméticas, usando árvores ordenadas enraizadas. Por exemplo, considere a representação de uma expressão aritmética que envolve as operações + (adição), − (subtração), ∗ (multiplicação), / (divisão) e ↑ (exponenciação). Usaremos parênteses para indicar a ordem das operações. Uma árvore enraizada ordenada pode ser usada para representar tais expressões, em que os vértices internos representam operações e as folhas representam as variáveis ou números. Cada operação opera em suas subárvores esquerda e direita (nesta ordem).

EXEMPLO 5 Qual é a árvore enraizada ordenada que representa a expressão $((x + y) \uparrow 2) + ((x - 4)/3)$?

Solução: A árvore binária para esta expressão pode ser construída de baixo para cima. Primeiro, é construída uma subárvore para $x + y$. A seguir, ela é incorporada como parte de uma subárvore maior que representa $(x + y) \uparrow 2$. Além disso, é construída uma subárvore para $x - 4$ e, a seguir, ela é incorporada à subárvore que representa $(x - 4)/3$. Finalmente, as subárvores que representam $(x + y) \uparrow 2$ e $(x - 4)/3$ são combinadas para formar a árvore enraizada ordenada que representa $((x + y) \uparrow 2) + ((x - 4)/3)$. Esses passos são mostrados na Figura 10. ◀

Um percurso em ordem simétrica da árvore binária que representa uma expressão produz a expressão original com os elementos e operações na mesma ordem que eles ocorreram originalmente, exceto pelas operações unárias, que em vez disso imediatamente seguem seus operandos. Por exemplo, todos os percursos em ordem simétrica das árvores binárias na Figura 11, que representam as expressões $(x + y)/(x + 3)$, $(x + (y/x)) + 3$ e $x + (y/(x + 3))$, levam à expressão infixa $x + y/x + 3$. Para tornar tais expressões não ambíguas, é necessário incluir parênteses no percurso em ordem simétrica sempre que encontrarmos uma operação. Dizemos que a expressão com todos os parênteses obtida dessa maneira está na **forma infixa**.

Obtemos a **forma prefixa** de uma expressão quando percorremos sua árvore enraizada em pré-ordem. Dizemos que as expressões escritas na ordem prefixa estão na **notação polonesa**, que é assim denominada em homenagem ao lógico polonês Jan Lukasiewicz. Uma expressão na notação prefixa (na qual cada operação tem um número especificado de operandos) não é ambígua,

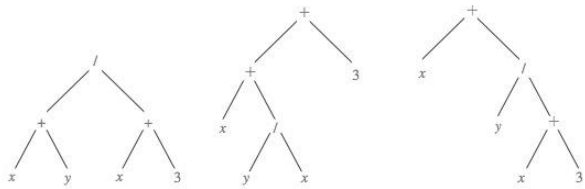


FIGURA 11 **Árvores Enraizadas que Representam $(x + y)/(x + 3)$, $(x + (y/x)) + 3$ e $x + (y/(x + 3))$.**

de modo que não é necessário nenhum parêntese nessa expressão. A verificação disso é deixada como um exercício para o leitor.

EXEMPLO 6 Qual é a forma prefixa para $((x + y) \uparrow 2) + ((x - 4)/3)$?

Solução: Obtemos a forma prefixa para esta expressão percorrendo a árvore binária que a representa, mostrada na Figura 10. Isso produz $+ \uparrow + x\, y\, 2\, / - x\, 4\, 3$. ◀

Na forma prefixa de uma expressão, um operador binário, tal como $+$, precede seus dois operandos. Portanto, podemos calcular uma expressão na forma prefixa trabalhando da direita para a esquerda. Quando encontramos um operador, efetuamos a operação correspondente com os dois operandos imediatamente à direita deste operando. Além disso, sempre que uma operação é efetuada, consideramos o resultado um novo operando.

EXEMPLO 7 Qual é o valor da operação prefixa $+ - * 2\, 3\, 5\, / \uparrow 2\, 3\, 4$?

Solução: Os passos usados para calcular esta expressão trabalhando da direita para a esquerda e efetuando as operações usando os operandos à direita são mostrados na Figura 12. O valor desta expressão é 3. ◀

Links

Obtemos a **forma pós-fixa** de uma expressão percorrendo sua árvore binária em pós-ordem. Dizemos que as expressões escritas na forma pós-fixa estão na **notação polonesa reversa**. Expressões na notação polonesa reversa não são ambíguas, de modo que os parênteses não são necessários. A verificação disto é deixada para o leitor.

EXEMPLO 8 Qual é a forma pós-fixa da expressão $((x + y) \uparrow 2) + ((x - 4)/3)$?

Solução: A forma pós-fixa da expressão é obtida seguindo um percurso em pós-ordem da árvore binária desta expressão, mostrado na Figura 10. Isso produz a expressão pós-fixa $x\, y + 2 \uparrow x\, 4 - 3\, / +$. ◀

Na forma pós-fixa de uma expressão, um operador binário segue seus dois operandos. Assim, para calcular uma expressão a partir de sua forma pós-fixa, trabalhe da esquerda para a direita, efetuando operações sempre que um operador vier a seguir de dois operandos. Depois de uma operação ser feita, o resultado dessa operação se torna um novo operando.

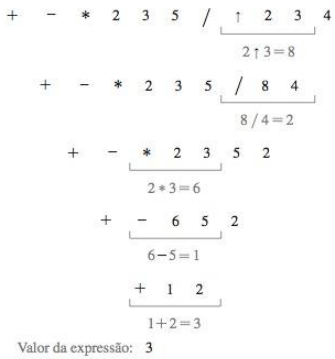


FIGURA 12 Calculando uma Expressão Prefixa.

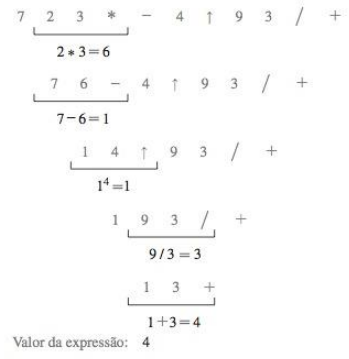


FIGURA 13 Calculando uma Expressão Pós-fixa.

EXEMPLO 9 Qual é o valor da expressão pós-fixa $7\ 2\ 3\ *\ -\ 4\ \uparrow\ 9\ 3\ /+$?

Solução: Os passos usados para calcular esta expressão começando da esquerda e efetuando as operações quando dois operandos são seguidos de um operador são mostrados na Figura 13. O valor dessa expressão é 4. ◀

As árvores enraizadas podem ser usadas para representar outros tipos de expressão, tal como as que representam proposições compostas e combinações de conjuntos. Nestes exemplos, ocorrem operadores unários, tal como a negação de uma proposição. Para representar tais operadores e seus operandos, são usados um vértice representando o operador e um filho deste vértice representando o operando.

EXEMPLO 10 Encontre a árvore enraizada ordenada que representa a proposição composta $(\neg(p \wedge q)) \leftrightarrow (\neg p \vee \neg q)$. A seguir, use esta árvore enraizada para encontrar as formas prefixa, pós-fixa e infixa desta expressão.

Links

JAN LUKASIEWICZ (1878–1956) Jan Lukasiewicz nasceu em uma família que falava polonês, em Lvov. Naquela época, Lvov era parte da Áustria, mas agora fica na Ucrânia. Seu pai era capitão no exército austríaco. Lukasiewicz se interessou pela matemática quando estava no ensino médio. Ele estudou matemática e filosofia na Universidade de Lvov, tanto na graduação quanto na pós-graduação. Depois de completar seu trabalho de doutorado, ele se tornou professor colaborador lá, e em 1911 foi nomeado professor efetivo. Quando a Universidade de Varsóvia foi reaberta como uma universidade polonesa em 1915, Lukasiewicz aceitou um convite para ingressar no seu corpo docente. Em 1919, ele foi ministro da educação polonês. Ele retornou à sua função de professor na Universidade de Varsóvia, onde permaneceu de 1920 a 1939, tendo sido reitor por duas vezes.

Lukasiewicz foi um dos co-fundadores da famosa Escola de Lógica de Varsóvia. Ele publicou seu famoso texto, *Elements of Mathematical Logic*, em 1928. Com sua influência, a lógica matemática se tornou uma disciplina obrigatória nos cursos de graduação em matemática e ciências na Polônia. Suas aulas eram consideradas excelentes, atraindo até mesmo estudantes da área de humanidades.

Lukasiewicz e sua esposa sofreram muito durante a Segunda Guerra Mundial, o que ele documentou em uma autobiografia publicada postumamente. Depois da guerra, eles viveram em exílio na Bélgica. Felizmente, em 1949 ele foi convidado para um cargo de professor na Royal Irish Academy, em Dublin.

Lukasiewicz trabalhou em lógica matemática por toda sua carreira. Seu trabalho em lógica a três valores foi uma contribuição importante no assunto. Entretanto, ele é mais conhecido nas comunidades matemáticas e de ciência da computação pela introdução da notação sem parênteses, atualmente conhecida como notação polonesa.

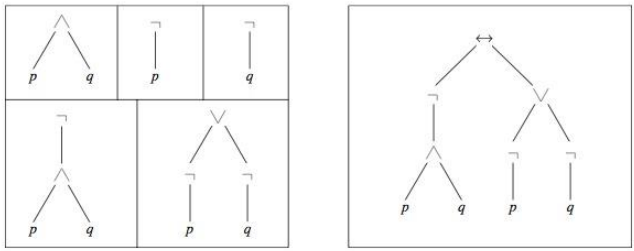


FIGURA 14 Construindo uma Árvore Enraizada para uma Proposição Composta.

Exemplos Extras

Solução: A árvore enraizada para esta proposição composta é construída de baixo para cima. Primeiro, são formadas as subárvores para $\neg p$ e $\neg q$ (em que $\neg$ é considerada uma operação unária). Além disso, é formada uma subárvore para $p \wedge q$. A seguir, são construídas as subárvores para $\neg (p \wedge q)$ e $(\neg p) \vee (\neg q)$. Finalmente, estas duas subárvores são usadas para formar a árvore enraizada final. Os passos deste procedimento são mostrados na Figura 14.

As formas prefixa, pós-fixa e infixa desta expressão são encontradas percorrendo esta árvore enraizada em pré-ordem, pós-ordem e ordem simétrica (incluindo parênteses), respectivamente. Estes percursos fornecem $\leftrightarrow \neg \wedge pq \vee \neg p \neg q, pq \wedge \neg p \neg q \neg \vee \leftrightarrow$ e $(\neg (p \wedge q)) \leftrightarrow ((\neg p) \vee (\neg q))$, respectivamente. ◀

Como as expressões prefixa e pós-fixa são não ambíguas e porque podem ser facilmente calculadas sem buscas para trás e para frente, elas são amplamente usadas em ciência da computação. Essas expressões são especialmente úteis na construção de compiladores.

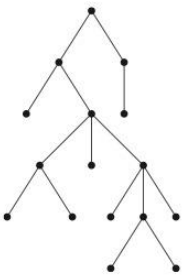
Exercícios

Nos exercícios 1 a 3, construa o sistema de endereçamento global para a árvore enraizada ordenada dada. A seguir, use isto para ordenar seus vértices utilizando a ordem lexicográfica de seus rótulos.

1.

2.

3.



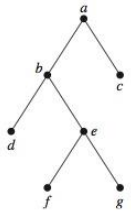
4. Suponha que o endereço de um vértice v na árvore enraizada ordenada T seja 3.4.5.2.4.
- **a)** Em que nível está v?
- **b)** Qual é o endereço do pai de v?
- **c)** Qual é o menor número de irmãos que v pode ter?
- **d)** Qual é o menor número possível de vértices em T se v tiver este endereço?
- **e)** Encontre os outros endereços que podem ocorrer.

5. Suponha que o vértice com maior endereço em uma árvore enraizada ordenada T tenha o endereço 2.3.4.3.1. É possível determinar o número de vértices de T?

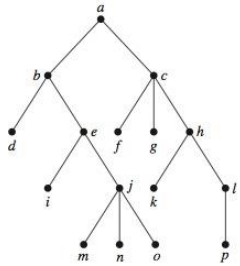
6. As folhas de uma árvore enraizada ordenada podem ter a seguinte lista de endereços globais? Em caso afirmativo, construa tal árvore enraizada ordenada.
a) 1.1.1, 1.1.2, 1.2, 2.1.1.1, 2.1.2, 2.1.3, 2.2, 3.1.1, 3.1.2.1, 3.1.2.2, 3.2
b) 1.1, 1.2.1, 1.2.2, 1.2.3, 2.1, 2.2.1, 2.3.1, 2.3.2, 2.4.2.1, 2.4.2.2, 3.1, 3.2.1, 3.2.2
c) 1.1, 1.2.1, 1.2.2, 1.2.2.1, 1.3, 1.4, 2, 3.1, 3.2, 4.1.1.1

Nos exercícios 7 a 9, determine a ordem na qual um percurso em pré-ordem visita os vértices da árvore enraizada ordenada dada.

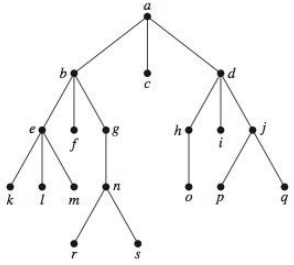
7.



8.



9.



10. Em qual ordem os vértices da árvore enraizada ordenada do Exercício 7 são visitados usando um percurso em ordem simétrica?

11. Em qual ordem os vértices da árvore enraizada ordenada do Exercício 8 são visitados usando um percurso em ordem simétrica?

12. Em qual ordem os vértices da árvore enraizada ordenada do Exercício 9 são visitados usando um percurso em ordem simétrica?

13. Em qual ordem os vértices da árvore enraizada ordenada do Exercício 7 são visitados usando um percurso em pós-ordem?

14. Em qual ordem os vértices da árvore enraizada ordenada do Exercício 8 são visitados usando um percurso em pós-ordem?

15. Em qual ordem os vértices da árvore enraizada ordenada do Exercício 9 são visitados usando um percurso em pós-ordem?

16. **a)** Represente a expressão $((x + 2) \uparrow 3) * (y - (3 + x)) - 5$ usando uma árvore binária.
Escreva esta expressão em
b) notação prefixa.
c) notação pós-fixa.
d) notação infixa.

17. **a)** Represente as expressões $(x + xy) + (x/y)$ e $x + ((xy + x)/y)$ usando árvores binárias.
Escreva estas expressões em
b) notação prefixa.
c) notação pós-fixa.
d) notação infixa.

18. **a)** Represente as proposições compostas $\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$ e $(\neg p \wedge (q \leftrightarrow \neg p)) \vee \neg q$ usando árvores enraizadas ordenadas.
Escreva estas expressões em
b) notação prefixa.
c) notação pós-fixa.
d) notação infixa.

19. **a)** Represente $(A \cap B) - (A \cup (B - A))$ usando uma árvore enraizada ordenada.
Escreva esta expressão em
b) notação prefixa.
c) notação pós-fixa.
d) notação infixa.

***20.** De quantas maneiras é possível colocar parênteses de modo completo na seqüência $\neg p \wedge q \leftrightarrow \neg p \vee \neg q$ para produzir uma expressão infixa?

***21.** De quantas maneiras é possível colocar parênteses de modo completo na seqüência $A \cap B - A \cap B - A$ para produzir uma expressão infixa?

22. Desenhe uma árvore enraizada ordenada que corresponda a cada uma das expressões aritméticas escritas na notação prefixa. A seguir, escreva cada expressão usando a notação infixa.
a) $+ * + - 5\ 3\ 2\ 1\ 4$
b) $\uparrow + 2\ 3 - 5\ 1$
c) $* / 9\ 3 + * 2\ 4 - 7\ 6$

23. Qual é o valor de cada uma destas expressões prefixas?
a) $- * 2 / 8\ 4\ 3$
b) $\uparrow - * 3\ 3 * 4\ 2\ 5$
c) $+ - \uparrow 3\ 2 \uparrow 2\ 3 / 6 - 4\ 2$
d) $* + 3 + 3 \uparrow 3 + 3\ 3\ 3$

24. Qual é o valor de cada uma destas expressões pós-fixas?
a) $5\ 2\ 1 - - 3\ 1\ 4 + + *$
b) $9\ 3 / 5 + 7\ 2 - *$
c) $3\ 2 * 2 \uparrow 5\ 3 - 8\ 4 / * -$

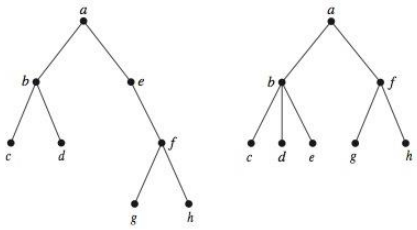
25. Construa a árvore enraizada ordenada cujo percurso em pré-ordem é $a, b, f, c, g, h, i, d, e, j, k, l$, em que a tem

quatro filhos, c tem três filhos, j tem dois filhos, b e e têm um filho cada e todos os outros vértices são folhas.

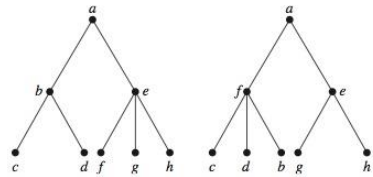
***26.** Mostre que uma árvore enraizada ordenada é univocamente determinada quando forem especificados uma lista de vértices gerada por um percurso em pré-ordem de uma árvore e o número de filhos de cada vértice.

***27.** Mostre que uma árvore enraizada ordenada é univocamente determinada quando forem especificados uma lista de vértices gerada por um percurso em pós-ordem de uma árvore e o número de filhos de cada vértice.

28. Mostre que os percursos em pré-ordem das duas árvores enraizadas ordenadas mostradas abaixo produzem a mesma lista de vértices. Observe que isso não contradiz a afirmação no Exercício 26, pois o número de filhos dos vértices internos das duas árvores enraizadas ordenadas é diferente.



29. Mostre que os percursos em pós-ordem das duas árvores enraizadas ordenadas produzem a mesma lista de vértices. Observe que isso não contradiz a afirmação no Exercício 27, pois o número de filhos dos vértices internos das duas árvores enraizadas ordenadas é diferente.



Fórmulas bem formadas na notação prefixa em um conjunto de símbolos e em um conjunto de operadores binários são definidas recursivamente por estas regras:

(*i*) se x for um símbolo, então x é uma fórmula bem formada na notação prefixa;

(*ii*) se X e Y forem fórmulas bem formadas e $*$ for um operador, então $*XY$ é uma fórmula bem formada.

30. Quais destas são fórmulas bem formadas no conjunto de símbolos $\{x, y, z\}$ e no conjunto de operadores binários $\{\times, +, \circ\}$?

a) $\times + + x y x$

b) $\circ x y \times x z$

c) $\times \circ x z \times \times x y$

d) $\times + \circ x x \circ x x x$

*** 31.** Mostre que qualquer fórmula bem formada na notação prefixa em um conjunto de símbolos e em um conjunto de operadores binários contém exatamente um símbolo a mais do que o número de operadores.

32. Dê uma definição de fórmulas bem formadas na notação pós-fixa em um conjunto de símbolos e em um conjunto de operadores binários.

33. Dê seis exemplos de fórmulas bem formadas com três ou mais operadores na notação pós-fixa no conjunto de símbolos $\{x, y, z\}$ e no conjunto de operadores $\{+, \times, \circ\}$.

34. Estenda a definição de fórmulas bem formadas na notação prefixa aos conjuntos de símbolos e operadores nos quais os operadores podem não ser binários.

10.4 Árvores Geradoras

Introdução

Considere o sistema de estradas no Maine representado pelo grafo simples mostrado na Figura 1(a). A única maneira pela qual as estradas podem ser mantidas abertas no inverno é desobstruindo-as da neve com freqüência. O departamento de estradas quer desobstruir o menor número possível de estradas, de modo que sempre haja estradas livres ligando duas cidades. Como isso pode ser feito?

Pelo menos cinco estradas precisam ser desobstruídas para garantir que exista um caminho entre duas cidades quaisquer. A Figura 1 (b) mostra esse conjunto de estradas. Observe que o subgrafo que representa essas estradas é uma árvore, pois é conexo e contém seis vértices e cinco arestas.

Este problema foi resolvido com um subgrafo conexo com um número mínimo de arestas que contém todos os vértices do grafo simples original. Tal grafo deve ser uma árvore.

DEFINIÇÃO 1 Seja G um grafo simples. Uma *árvore geradora* de G é um subgrafo de G, que é uma árvore que contém todos os vértices de G.

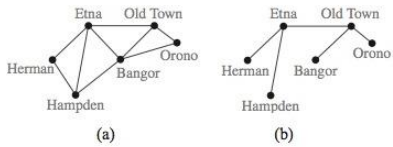


FIGURA 1 (a) Um Sistema de Estradas e (b) um Conjunto de Estradas a Serem Desobstruídas.

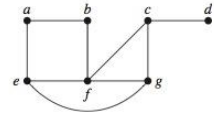


FIGURA 2 O Grafo Simples *G*.

Um grafo simples com uma árvore geradora deve ser conexo, pois existe um caminho na árvore geradora entre quaisquer dois vértices. A recíproca também é verdadeira; ou seja, todo grafo simples conexo tem uma árvore geradora. Daremos um exemplo antes de demonstrar este resultado.

EXEMPLO 1 Encontre uma árvore geradora do grafo simples G mostrado na Figura 2.

Solução: O grafo G é conexo, mas não é uma árvore, pois contém ciclos simples. Remova a aresta $\{a, e\}$. Isso elimina um ciclo simples e o subgrafo resultante ainda é conexo e ainda contém todos os vértices de G. A seguir, remova a aresta $\{e, f\}$ para eliminar um segundo ciclo simples. Finalmente, remova a aresta $\{c, g\}$ para produzir um grafo simples sem nenhum ciclo simples. Este subgrafo é uma árvore geradora, pois é uma árvore que contém todos os vértices de G. A seqüência de remoção de arestas usada para produzir a árvore geradora está ilustrada na Figura 3.

A árvore mostrada na Figura 3 não é a única árvore geradora de G. Por exemplo, cada uma das árvores mostradas na Figura 4 é uma árvore geradora de G. ◀

TEOREMA 1 Um grafo simples é conexo se e somente se ele tem uma árvore geradora.

Demonstração: Primeiro, suponha que um grafo simples G tenha uma árvore geradora T. T contém todos os vértices de G. Além disso, existe um caminho em T entre quaisquer dois de seus vértices. Como T é um subgrafo de G, existe um caminho em G entre quaisquer dois de seus vértices. Portanto, G é conexo.

Suponha agora que G seja conexo. Se G não for uma árvore, ele deve conter um ciclo simples. Remova uma aresta de um destes ciclos simples. O subgrafo resultante tem uma aresta a menos, mas ainda contém todos os vértices de G e é conexo. Este subgrafo ainda é conexo porque quando dois vértices são ligados por um caminho que contém a aresta removida, eles são ligados por um caminho que não contém esta aresta. Podemos construir esse caminho inserindo no caminho original, no ponto em que a aresta removida estava, o ciclo simples com esta aresta removi-

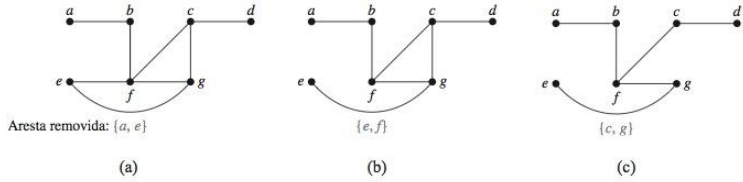


FIGURA 3 Produção de uma Árvore Geradora de *G* pela Remoção de Arestas que Formam Ciclos Simples.

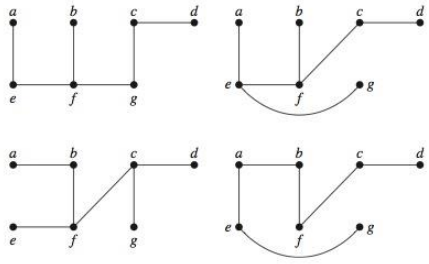


FIGURA 4 Árvores Geradoras de *G*.

da. Se este subgrafo não for uma árvore, existe um ciclo simples; do mesmo modo como anteriormente, remova uma aresta que esteja em um ciclo simples. Repita este processo até que não sobre nenhum ciclo simples. Isso é possível porque existe apenas um número finito de arestas no grafo. Este processo termina quando não restar mais nenhum ciclo. Uma árvore é produzida, pois o grafo permanece conexo conforme as arestas são removidas. Esta árvore é uma árvore geradora porque contém todos os vértices de *G*. ◁

Árvores geradoras são importantes em redes de comunicação de dados, como o Exemplo 2 mostra.

EXEMPLO 2 **Distribuição Múltipla em IP** Árvores geradoras desempenham papel importante em distribuição múltipla em redes de Internet Protocol (IP). Para mandar dados de um computador fonte para computadores recipientes múltiplos, cada um dos quais é uma sub-rede, dados poderiam ser mandados separadamente para cada computador. Este tipo de rede de comunicação, chamada de distribuição única (*unicasting*), é ineficiente, pois muitas cópias dos mesmos dados são transmitidas pela rede. Para tornar a transmissão de dados para computadores recipientes múltiplos mais eficiente, é usada transmissão múltipla em IP. Com transmissão múltipla em IP, um computador manda uma única cópia dos dados pela rede, e, conforme os dados atingem routers intermediários, são encaminhados para um ou mais outros routers, de modo que finalmente todos os computadores recipientes em suas várias sub-redes recebem esses dados. (Routers são computadores destinados a encaminhar um pacote de dados em IP entre sub-redes em uma rede. Na distribuição múltipla, os routers usam endereços de Classe D, cada um representando uma seção que os computadores recipientes podem acessar; veja o Exemplo 16 na Seção 5.1.)

Links

Para que os dados atinjam os computadores recipientes tão rapidamente quanto possível, não deve haver nenhum laço (o que na terminologia de teoria dos grafos são ciclos) no caminho que os dados seguem através da rede. Ou seja, uma vez que os dados tenham atingido um router particular, eles nunca deveriam retornar a este router. Para evitar laços, os routers de distribuição múltipla usam algoritmos de rede para construir uma árvore geradora no grafo que tem a fonte de distribuição múltipla, os roteadores e as sub-redes com computadores recipientes como vértices, com as arestas representando as conexões entre computadores e/ou routers. A raiz desta árvore geradora é a fonte de distribuição múltipla. As sub-redes com os computadores recipientes são folhas da árvore. (Observe que as sub-redes de comunicação que não contêm estações recipientes não são incluídas no grafo.) Isso é ilustrado na Figura 5. ◀

Busca em Profundidade

Links

A demonstração do Teorema 1 dá um algoritmo para encontrar árvores geradoras pela remoção de arestas de ciclos simples. Este algoritmo é ineficiente, pois exige que ciclos simples sejam identificados. Em vez de construir árvores geradoras removendo arestas, as árvores gera-

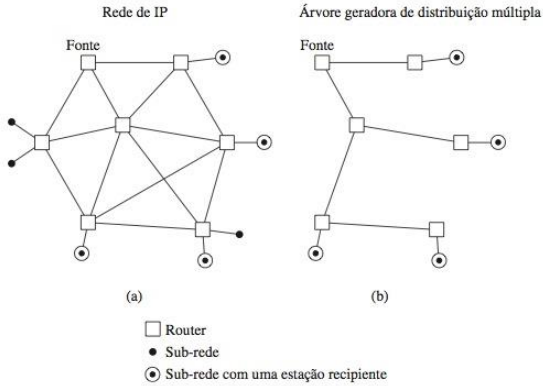


FIGURA 5 **Uma Árvore Geradora de Distribuição Múltipla.**

doras podem ser construídas pela adição sucessiva de arestas. Dois algoritmos com base neste princípio serão apresentados aqui.

Demo

Podemos construir uma árvore geradora para um grafo simples conexo usando **busca em profundidade**. Vamos formar uma árvore enraizada, e a árvore geradora será o grafo não orientado subjacente a esta árvore enraizada. Escolha arbitrariamente um vértice do grafo como raiz. Forme um caminho começando neste vértice adicionando sucessivamente vértices e arestas, em que cada nova aresta é incidente ao último vértice do caminho e a um vértice que ainda não esteja no caminho. Continue adicionando vértices e arestas a este caminho enquanto for possível. Se o caminho passar por todos os vértices do grafo, a árvore que consiste neste caminho é uma árvore geradora. Entretanto, se o caminho não passar por todos os vértices, mais vértices e mais arestas devem ser adicionados. Volte para o penúltimo vértice no caminho e, se possível, forme um novo caminho começando neste vértice e passando por vértices que ainda não tenham sido visitados. Se isso não puder ser feito, volte mais um vértice no caminho, ou seja, o antepenúltimo vértice, e tente novamente.

Repita este procedimento, começando do último vértice visitado, se movendo para trás no caminho, um vértice de cada vez, formando novos caminhos que sejam tão longos quanto possível até que não possa ser adicionada mais nenhuma aresta. Como o grafo tem um número finito de arestas e é conexo, este processo acaba com a produção de uma árvore geradora. Cada vértice que termina um caminho em um estágio do algoritmo será uma folha na árvore enraizada, e cada vértice em que um caminho que começa neste vértice é construído será um vértice interno.

O leitor deve observar a natureza recursiva deste procedimento. Observe também que, se os vértices do grafo estiverem ordenados, as escolhas das arestas em cada estágio do procedimento estão todas determinadas quando sempre escolhemos o primeiro vértice na ordenação que esteja disponível. Entretanto, nem sempre ordenaremos explicitamente os vértices de um grafo.

A busca em profundidade é também chamada de **backtraking** (retroceder), pois o algoritmo retorna a vértices previamente visitados para adicionar caminhos. O Exemplo 3 ilustra o *backtracking*.

EXEMPLO 3 Use a busca em profundidade para encontrar uma árvore geradora para o grafo G mostrado na Figura 6.

Exemplos Extras

Solução: Os passos usados na busca em profundidade para produzir uma árvore geradora de G estão mostrados na Figura 7. Começamos arbitrariamente com o vértice f. É construído um caminho adicionando sucessivamente arestas incidentes em vértices que ainda não estejam no caminho, enquanto isso for possível. Isso produz um caminho f, g, h, k, j (observe que outros caminhos poderiam ter sido construídos). A seguir, retroceda para k. Não existe nenhum caminho começando

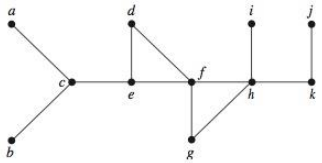


FIGURA 6 O Grafo *G*.

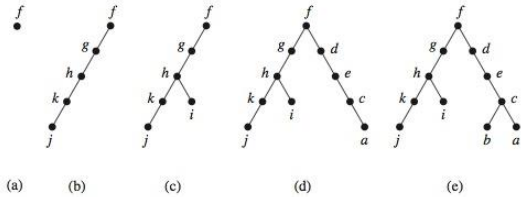


FIGURA 7 Busca em Profundidade de *G*.

em *k* que contenha vértices que ainda não tenham sido visitados. Assim, retrocedemos para *h*. Forme o caminho *h*, *i*. A seguir, retroceda para *h* e depois para *f*. A partir de *f* construímos o caminho *f*, *d*, *e*, *c*, *a*. Então, retrocedemos para *c* e formamos o caminho *c*, *b*. Isso produz uma árvore geradora. ◀

As arestas escolhidas pela busca em profundidade de um grafo são chamadas **arestas da árvore**. Todas as outras arestas do grafo devem conectar um vértice a um ancestral ou descendente deste vértice na árvore. Estas arestas são chamadas de **arestas de retorno**. (O Exercício 39 pede uma demonstração deste fato.)

EXEMPLO 4 Na Figura 8, destacamos as arestas da árvore encontrada pela busca em profundidade que começa no vértice *f* mostrando-as com linhas cinza. As arestas de retorno (e, f) e (f, h) são mostradas com linhas pretas mais finas. ◀

Explicamos como encontrar uma árvore geradora de um grafo usando busca em profundidade. Entretanto, nossa discussão até agora não evidenciou a natureza recursiva da busca em profundidade. Para ajudar a tornar clara a natureza recursiva do algoritmo, precisamos de um pouco de terminologia. Dizemos que *exploramos* a partir de um vértice v quando executamos os passos da busca em profundidade, começando quando v é adicionado à árvore e terminando quanto tivermos retrocedido para v pela última vez. A observação-chave necessária para entender a natureza recursiva do algoritmo é que, quando adicionamos uma aresta que conecta um vértice v a um vértice w, terminamos a exploração a partir de w antes de retornar a v para completar a exploração a partir de v.

No Algoritmo 1, construímos a árvore geradora de um grafo G com vértices $v_1, ..., v_n$ escolhendo primeiro o vértice v_1 para ser a raiz. Inicialmente, supomos T como a árvore com apenas este vértice. A cada passo, adicionamos um novo vértice à árvore T juntamente com uma aresta de um vértice já em T a este novo vértice e exploramos a partir deste novo vértice. Observe que, ao completar o algoritmo, T não conterá nenhum ciclo simples, pois não é adicionada nenhuma aresta que conecte um vértice que já esteja na árvore. Além disso, T permanece conexa enquanto é construída. (Estas duas últimas observações podem ser facilmente demonstradas com indução matemática.) Como G é conexo, todo vértice em G é visitado pelo algoritmo e adicionado à árvore (como o leitor deve verificar). Segue que T é uma árvore geradora de G.

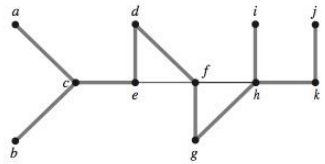


FIGURA 8 As Arestas da Árvore e as Arestas de Retorno da Busca em Profundidade do Exemplo 4.

ALGORITMO 1 Busca em Profundidade.

procedure *DFS*(*G*: grafo conexo com vértices $v_1, v_2, ..., v_n$)
$T :=$ árvore que consiste apenas no vértice v_1
visit (v_1)
procedure *visit*(v: vértice de G)
for cada vértice w adjacente a v e ainda não em T
begin
  adicione vértice w e aresta $\{v, w\}$ a T
  visit(w)
end

Analisamos agora a complexidade computacional do algoritmo da busca em profundidade. A observação-chave é que, de cada vértice v, o procedimento *visit*(v) é chamado quando o vértice v é encontrado pela primeira vez em uma busca e não é chamado novamente. Supondo que as listas de adjacência de G estejam disponíveis (veja a Seção 9.3), não é necessário nenhum cálculo para encontrar os vértices adjacentes a v. Conforme seguimos os passos do algoritmo, examinamos cada aresta no máximo duas vezes para determinar se adicionamos esta aresta e uma de suas extremidades à árvore. Conseqüentemente, o procedimento *DFS* constrói uma árvore geradora usando $O(e)$, ou $O(n^2)$, passos em que e e n são o número de arestas e o de vértices em G, respectivamente. [Observe que um passo envolve o exame de um vértice para ver se ele já está na árvore geradora quando ela está sendo construída e a adição deste vértice e da aresta correspondente, se ele ainda não estiver na árvore. Usamos também a desigualdade $e \leq n(n-1)/2$, que é válida para qualquer grafo simples.]

A busca em profundidade pode ser usada como base para algoritmos que resolvem muitos problemas diferentes. Por exemplo, pode ser usada para encontrar caminhos e ciclos em um grafo, pode ser usada para determinar as componentes conexas de um grafo e pode ser usada para determinar vértices de corte de um grafo conexo. Como veremos, a busca em profundidade é a base das técnicas de *backtracking* usadas na busca de soluções para problemas computacionalmente difíceis. (Ver [GrYe99], [Ma89] e [CoLeRiSt01] para uma discussão dos algoritmos com base em busca em profundidade.)

Busca em Largura

Demo

Links

Podemos também produzir uma árvore geradora de um grafo simples pelo uso da **busca em largura**. Novamente, será construída uma árvore enraizada, e o grafo não orientado subjacente a esta árvore enraizada é a árvore geradora. Escolha arbitrariamente uma raiz entre os vértices do grafo. Então, adicione todas as arestas incidentes a este vértice. Os novos vértices adicionados neste estágio se tornarão os vértices no nível 1 da árvore geradora. Ordene-os arbitrariamente. A seguir, para cada vértice no nível 1, visitados em ordem, adicione cada aresta incidente a este vértice à árvore enquanto isso não produzir um ciclo simples. Ordene arbitrariamente os filhos de cada vértice do nível 1. Isso produz os vértices de nível 2 na árvore. Siga o mesmo procedimento até que todos os vértices da árvore tenham sido adicionados. O procedimento termina, pois existe apenas um número finito de arestas no grafo. É produzida uma árvore geradora porque produzimos uma árvore que contém todos os vértices do grafo. Uma ilustração da busca em largura é dada no Exemplo 5.

EXEMPLO 5 Use busca em largura para encontrar uma árvore geradora para o grafo mostrado na Figura 9.

Exemplos Extras

Solução: Os passos da busca em largura estão mostrados na Figura 10. Escolhemos o vértice e para ser a raiz. Então, adicionamos as arestas incidentes a todos os vértices adjacentes a e, de modo que as arestas de e a b, a d, a f e a i são adicionadas. Estes quatro vértices estão no nível 1 da árvore. A seguir, adicionamos as arestas destes vértices no nível 1 a vértices adjacentes que

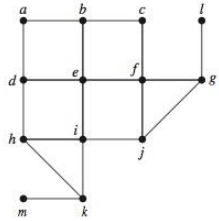


FIGURA 9 **Um Grafo *G*.**

ainda não estejam na árvore. Portanto, as arestas de b a a e a c são adicionadas, bem como as arestas de d a h, de f a j e a g, e de i a k. Os novos vértices a, c, h, j, g e k estão no nível 2. A seguir, adicione arestas destes vértices aos vértices adjacentes que ainda não estejam no grafo. Isso adiciona as arestas de g a l e de k a m. ◀

Descrevemos a busca em largura em pseudocódigo no Algoritmo 2. Neste algoritmo, supomos que os vértices do grafo conexo G estão ordenados como $v_1, v_2, \ldots, v_n$. No algoritmo usamos o termo "processo" para descrever o procedimento de adicionar novos vértices, e as arestas correspondentes, à árvore adjacente ao vértice que está sendo processado no momento enquanto nenhum ciclo fechado for produzido.

ALGORITMO 2 **Busca em Largura.**

procedure *BFS*(G: grafo conexo com vértices $v_1, v_2, \ldots, v_n$)
T := árvore que consiste apenas no vértice v_1
L := lista vazia
ponha v_1 na lista L dos vértices não processados
while L não estiver vazia
begin
  remova o primeiro vértice, v, de L
  for cada vizinho w de v
    if w não estiver em L e não estiver em T **then**
    begin
      adicione w ao final da lista L
      adicione w e a aresta $\{v, w\}$ a T
    end
end

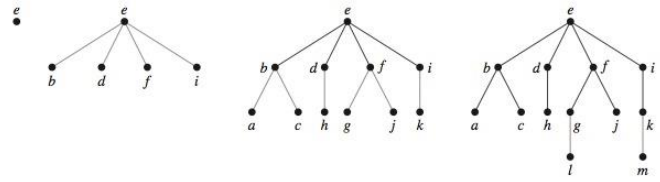


FIGURA 10 **Busca em Largura de *G*.**

Agora analisamos a complexidade computacional da busca em largura. Para cada vértice v no grafo, examinamos todos os vértices adjacentes a v e adicionamos cada vértice que ainda não tenha sido visitado à árvore T. Supondo que as listas de adjacência ao grafo estejam disponíveis, nenhum cálculo é necessário para determinar quais vértices são adjacentes a um dado vértice. Como na análise do algoritmo de busca em profundidade, vemos que examinamos cada aresta no máximo duas vezes para determinar se deveríamos adicionar esta aresta e sua extremidade que ainda não está na árvore. Segue que o algoritmo de busca em largura usa $O(e)$ ou $O(n^2)$ passos.

Aplicações de *Backtracking*

Existem problemas que só podem ser resolvidos fazendo uma busca exaustiva de todas as soluções possíveis. Uma maneira de procurar sistematicamente por uma solução é usar uma árvore de decisão, em que cada vértice interno representa uma decisão e cada folha, uma solução possível. Para encontrar uma solução via *backtracking*, primeiro faça uma seqüência de decisões em uma tentativa de encontrar uma solução enquanto isso for possível. A seqüência de decisões pode ser representada por um caminho na árvore de decisão. Uma vez que se saiba que nenhuma solução pode resultar de uma seqüência mais longa de decisões, retroceda ao pai do vértice em análise e trabalhe em direção a uma solução com uma outra seqüência de decisões, se isso for possível. O procedimento continua até que seja encontrada uma solução ou até que seja estabelecido que não existe solução. Os exemplos 6 a 8 ilustram a utilidade do *backtracking*.

EXEMPLO 6 **Coloração de grafos** Como o *backtracking* pode ser usado para decidir se um grafo pode ser colorido usando n cores?

Solução: Podemos resolver este problema usando *backtracking* da seguinte maneira. Primeiro, escolha um vértice a e associe a ele a cor 1. Então, escolha um segundo vértice b, e se b não for adjacente a a, associe a ele a cor 1. Caso contrário, associe a cor 2 a b. A seguir, vá para um terceiro vértice c. Use a cor 1, se possível, para c. Caso contrário, use a cor 2, se possível. Apenas se nem a cor 1 nem a cor 2 puderem ser usadas a cor 3 deverá ser usada. Continue este processo enquanto for possível associar uma das n cores a cada vértice adicional, sempre usando a primeira cor permitida da lista. Se for atingido um vértice que não pode ser colorido por nenhuma das n cores, retroceda à última associação feita e mude a coloração do último vértice colorido, se possível, usando a próxima cor disponível na lista. Se não for possível mudar esta coloração, retroceda mais para associações prévias, um passo para trás de cada vez, até que seja possível mudar a cor de um vértice. Então, continue a associar as cores dos vértices adicionais enquanto for possível. Se uma coloração usando n cores existir, o *backtracking* a produzirá. (Infelizmente, este procedimento pode ser extremamente ineficiente.)

Em particular, considere o problema de colorir o grafo mostrado na Figura 11 com três cores. A árvore mostrada na Figura 11 ilustra como o *backtracking* pode ser usado para construir uma 3-coloração. Neste procedimento, o vermelho será usado primeiro, depois o azul e, finalmente, o verde. Este exemplo simples pode obviamente ser feito sem *backtracking*, mas é uma boa ilustração da técnica.

Nesta árvore, o caminho inicial da raiz, que representa a associação de vermelho a a, leva a uma coloração com a vermelho, b azul, c vermelho e d verde. É impossível colorir e usando qualquer uma das três cores quando a, b, c e d estão coloridos desta forma. Assim, retroceda para o pai do vértice que representa esta coloração. Como nenhuma outra cor pode ser usada para d, retroceda mais um nível. Então, mude a cor de c para verde. Obtemos uma coloração do grafo associando vermelho a d e verde a e. ◀

EXEMPLO 7 **O Problema das *n* Rainhas** O problema das n rainhas pergunta como n rainhas podem ser colocadas em um tabuleiro de xadrez $n \times n$, de modo que nenhum par de rainhas possa atacar uma a outra. Como o *backtracking* pode ser usado para resolver o problema das n rainhas?

Links

Solução: Para resolver este problema, devemos encontrar n posições em um tabuleiro $n \times n$, de modo que duas destas posições nunca estejam na mesma linha, na mesma coluna ou na mesma diagonal [uma diagonal consiste em todas as posições (i, j), com $i + j = m$ para algum m ou $i - j = m$ para algum m]. Usaremos *backtracking* para resolver o problema das n rainhas. Come-

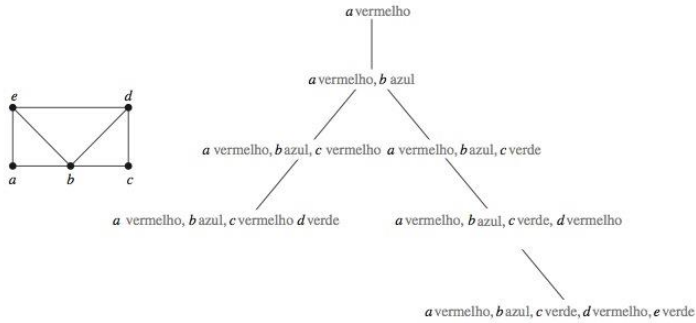


FIGURA 11 Coloração de um Grafo Usando *Backtracking*.

çamos com um tabuleiro vazio. No estágio $k + 1$, tentamos colocar uma rainha adicional no tabuleiro, na coluna $(k + 1)$, em que já há rainhas nas primeiras k colunas. Examinamos quadrados na coluna $(k + 1)$, começando com o quadrado na primeira linha, procurando uma posição para colocar esta rainha de modo que ela não esteja na mesma linha ou na mesma diagonal de uma rainha que já esteja no tabuleiro. (Já sabemos que ela não está na mesma coluna.) Se for impossível encontrar uma posição para colocar a rainha na coluna $(k + 1)$, retroceda para a colocação da rainha na coluna k e coloque esta rainha na próxima linha permitida nesta coluna, se tal linha existir. Se não existir tal linha, retroceda mais.

Em particular, a Figura 12 mostra uma solução por *backtracking* para o problema das quatro rainhas. Nesta solução, colocamos uma rainha na primeira linha da primeira coluna. A seguir, colocamos uma rainha na terceira linha da segunda coluna. Entretanto, isso torna impossível colocar uma rainha na terceira coluna. Assim, retrocedemos e colocamos uma rainha na quarta linha da segunda coluna. Quando fazemos isso, podemos colocar uma rainha na segunda linha da terceira coluna. Contudo, não há nenhuma maneira de adicionar uma rainha à quarta coluna. Isso mostra que não achamos nenhuma solução quando uma rainha é colocada na primeira linha da primeira coluna. Retrocedemos para o tabuleiro vazio e colocamos uma rainha na segunda linha da primeira coluna. Isso nos leva à solução mostrada na Figura 12. ◀

EXEMPLO 8 **Somas de Subconjuntos** Considere este problema. Dado um conjunto de inteiros positivos x_1, x_2, ..., x_n, encontre um subconjunto deste conjunto de inteiros que tenha M como sua soma. Como o *backtracking* pode ser usado para resolver este problema?

Solução: Começamos com uma soma sem nenhum termo. Construímos a soma adicionando termos sucessivamente. Um inteiro na seqüência é incluído se a soma permanecer menor do que M quando este inteiro for adicionado à soma. Se for atingida uma soma tal que a adição de qualquer termo é maior do que M, retroceda excluindo o último termo da soma.

A Figura 13 mostra uma solução de *backtracking* para o problema de encontrar um subconjunto de {31, 27, 15, 11, 7, 5} com a soma igual a 39. ◀

Busca em Profundidade em Grafos Orientados

Podemos modificar facilmente tanto a busca em profundidade quanto a busca em largura, de modo que elas possam ser executadas quando for dado um grafo orientado como entrada. Entretanto, a saída não será necessariamente uma árvore geradora, mas, em vez disso, uma floresta geradora. Em ambos os algoritmos, podemos adicionar uma aresta apenas quando ela estiver

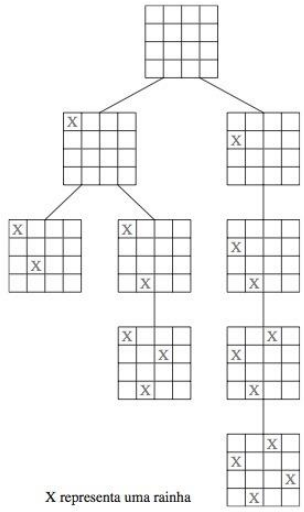


FIGURA 12 Uma Solução de *Backtracking* para o Problema das Quatro Rainhas.

orientada para longe do vértice que está sendo visitado e para um vértice que ainda não foi adicionado. Se em um estágio de qualquer um dos algoritmos descobrirmos que não existe nenhuma aresta começando em um vértice já adicionado para um que ainda não tenha sido adicionado, o próximo vértice adicionado por este algoritmo se torna a raiz de uma nova árvore na floresta geradora. Isso é ilustrado no Exemplo 9.

EXEMPLO 9 Qual é a saída da busca em profundidade, dado o grafo G mostrado na Figura 14(a) como entrada?

Solução: Começamos a busca em profundidade no vértice a e adicionamos os vértices b, c e g e as arestas correspondentes, onde ficamos bloqueados. Retrocedemos para c, mas ainda ficamos

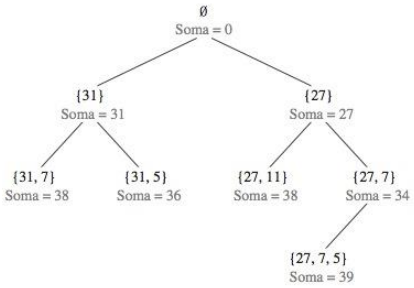


FIGURA 13 Encontrar uma Soma Igual a 39 Usando *Backtracking*.

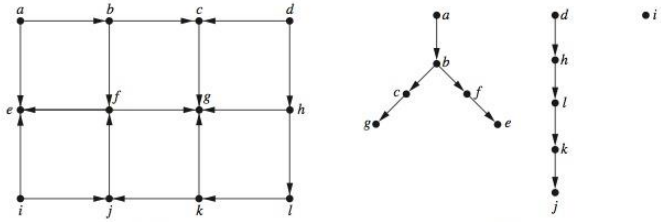


FIGURA 14 Busca em Profundidade de um Grafo Orientado.

bloqueados, e então retrocedemos para *b*, em que adicionamos os vértices *f* e *e* e as arestas correspondentes. *Backtracking* nos leva de volta todo o caminho até *a*. Então começamos uma nova árvore em *d* e adicionamos os vértices *h*, *l*, *k* e *j* e as arestas correspondentes. Retrocedemos para *k* e depois para *l*, para *h* e de volta para *d*. Finalmente, começamos uma nova árvore em *i*, completando a busca em profundidade. A saída é mostrada na Figura 14(b). ◀

A busca em profundidade em grafos orientados é a base de muitos algoritmos (ver [GrYe99], [Ma89] e [CoLeRiSt01]). Pode ser usada para determinar se um grafo orientado tem um ciclo, pode ser usada para obter uma ordenação topológica de um grafo e pode ser usada para encontrar as componentes fortemente conexas de um grafo orientado.

Concluímos esta seção com uma aplicação da busca em profundidade e da busca em largura para instrumentos de busca na Web.

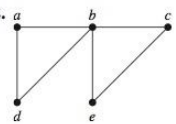
EXEMPLO 10 ***Web Spiders*** Para indexar páginas da Web, mecanismos de busca, como Google e Yahoo, exploram sistematicamente a Web a partir de páginas conhecidas. Esses mecanismos de busca usam programas chamados de *Web spiders* (ou *crawlers* ou *bots**) para visitar páginas da Web e analisar seu conteúdo. Os *Web spiders* usam tanto busca em profundidade quanto busca em largura para criar índices. Como descrito no Exemplo 8 da Seção 9.1, as páginas da Web e os links entre elas podem ser modelados por um grafo orientado chamado de grafo da Web. As páginas da Web são representadas por vértices e os links são representados por arestas orientadas. Usando busca em profundidade, é selecionada uma página inicial da Web, um link é seguido por uma segunda página da Web (se existir esse link), um link na segunda página da Web é seguido por uma terceira página da Web, se existir esse link, e assim por diante, até que seja encontrada uma página sem novos links. O *backtracking* é então utilizado para examinar links em níveis anteriores e procurar novos links, e assim por diante. (Por causa de limitações práticas, os *Web spiders* têm limitantes para a profundidade que eles investigam na busca em profundidade.) Usando busca em largura, é selecionada uma página inicial da Web e um link nesta página é seguido por uma segunda página da Web, então, um segundo link na página inicial é seguido (se existir), e assim por diante, até que todos os links da página inicial tenham sido seguidos. Desse modo, os links das páginas no nível 1 são seguidos, página por página, e assim por diante. ◀

Exercícios

1. Quantas arestas devem ser removidas de um grafo conexo com *n* vértices e *m* arestas para produzir uma árvore geradora?

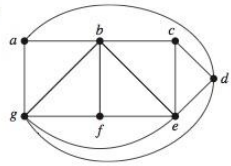
Nos exercícios 2 a 6, encontre a árvore geradora para o grafo mostrado removendo arestas em ciclos simples.

2.

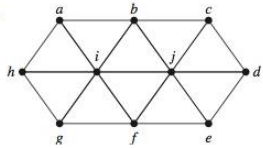


* N.T.: Respectivamente, "que se arrasta pelo chão" e "berne".

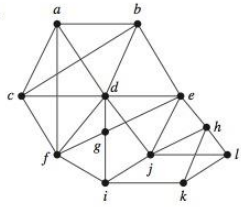
3.



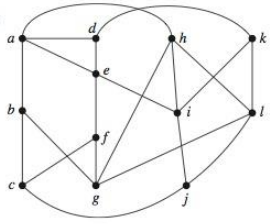
4.



5.



6.

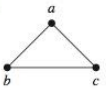


7. Encontre uma árvore geradora para cada um destes grafos.

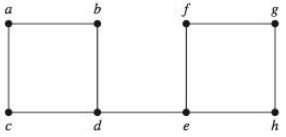
a) K_5 **b)** $K_{4,4}$ **c)** $K_{1,6}$
d) Q_3 **e)** C_5 **f)** W_5

Nos exercícios 8 a 10, desenhe todas as árvores geradoras dos grafos simples dados.

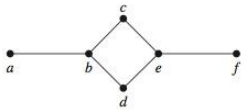
8.



9.



10.



***11.** Quantas árvores geradoras diferentes cada um destes grafos simples têm?

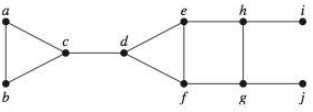
a) K_3 **b)** K_4 **c)** $K_{2,2}$ **d)** C_5

***12.** Quantas árvores geradoras não isomorfas cada um destes grafos simples têm?

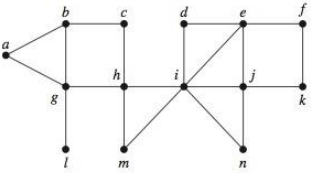
a) K_3 **b)** K_4 **c)** K_5

Nos exercícios 13 a 15, use busca em profundidade para produzir uma árvore geradora para o grafo simples dado. Escolha a como a raiz da árvore geradora e suponha que os vértices estão ordenados alfabeticamente.

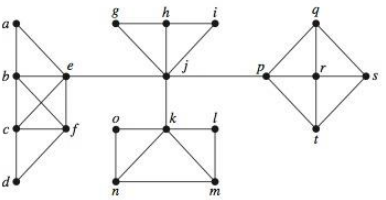
13.



14.



15.



16. Use busca em largura para produzir uma árvore geradora para cada um dos grafos simples dos exercícios 13 a 15. Escolha a como a raiz de cada árvore geradora.

17. Use busca em profundidade para encontrar uma árvore geradora de cada um destes grafos.

a) W_6 (veja o Exemplo 7 da Seção 9.2), começando no vértice de grau 6.

b) K_5

c) $K_{3,4}$, começando no vértice de grau 3.

d) Q_3

18. Use busca em largura para encontrar uma árvore geradora para cada um dos grafos do Exercício 17.

19. Descreva as árvores produzidas pela busca em largura e pela busca em profundidade do grafo roda W_n, começando

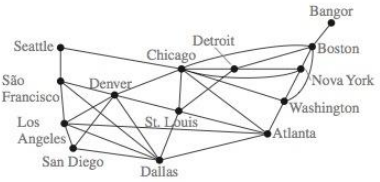
no vértice de grau n, em que n é um inteiro com $n \geq 3$. (Veja o Exemplo 7 da Seção 9.2.) Justifique suas respostas.

20. Descreva as árvores produzidas pela busca em largura e pela busca em profundidade do grafo completo K_n, em que n é um inteiro positivo. Justifique suas respostas.

21. Descreva as árvores produzidas pela busca em largura e pela busca em profundidade do grafo bipartido completo $K_{m,n}$, começando em um vértice de grau m, em que m e n são inteiros positivos. Justifique suas respostas.

22. Explique como a busca em largura e a busca em profundidade podem ser usadas para determinar se um grafo é bipartido.

23. Suponha que uma companhia aérea precise reduzir sua escala de vôos para economizar dinheiro. Se suas rotas originais forem como as ilustradas aqui, quais vôos poderiam ser suspensos para manter o serviço entre todos os pares de cidades (em que pode ser necessário combinar vôos para voar de uma cidade para outra)?



24. Quando uma aresta de um grafo simples conexo precisa estar em todas as árvores geradoras desse grafo?

25. Quando um grafo simples conexo tem exatamente uma árvore geradora?

26. Explique como a busca em largura e a busca em profundidade podem ser usadas para ordenar os vértices de um grafo conexo.

***27.** Mostre que o comprimento do caminho mais curto entre os vértices v e u em um grafo simples conexo é igual ao número do nível de u na árvore geradora por largura de G com raiz em v.

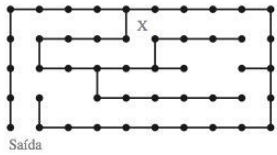
28. Use o *backtracking* para tentar encontrar uma coloração de cada um dos grafos dos exercícios 7 a 9 da Seção 9.8 usando três cores.

29. Use o *backtracking* para resolver o problema das n rainhas para estes valores de n.
a) $n = 3$ **b)** $n = 5$ **c)** $n = 6$

30. Use o *backtracking* para encontrar um subconjunto, se existir, do conjunto $\{27, 24, 19, 14, 11, 8\}$ com soma
a) 20. **b)** 41. **c)** 60.

31. Explique como o *backtracking* pode ser usado para encontrar um caminho ou ciclo hamiltoniano em um grafo.

32. **a)** Explique como o *backtracking* pode ser usado para encontrar um caminho para fora de um labirinto, dadas a posição inicial e a posição de saída. Considere o labirinto dividido em posições, em que em cada posição o conjunto de movimentos disponíveis inclui de uma a quatro possibilidades (para cima, para baixo, para a direita e para a esquerda).
b) Encontre um caminho da posição inicial marcada por X até a saída do labirinto.



Uma **floresta geradora** de um grafo G é uma floresta que contém todos os vértices de G tal que dois vértices estão na mesma árvore da floresta quando existir um caminho em G entre esses dois vértices.

33. Mostre que todo grafo simples finito tem uma floresta geradora.

34. Quantas árvores estão na floresta geradora de um grafo?

35. Quantas arestas devem ser removidas para produzir uma floresta geradora de um grafo com n vértices, m arestas e c componentes conexas?

36. Desenvolva um algoritmo para construir uma floresta geradora de um grafo baseado em excluir arestas que formam ciclos simples.

37. Desenvolva um algoritmo para construir uma floresta geradora de um grafo baseado em busca em profundidade.

38. Desenvolva um algoritmo para construir uma floresta geradora de um grafo baseado em busca em largura.

39. Seja G um grafo conexo. Mostre que, se T for uma árvore geradora de G construída usando busca em profundidade, então uma aresta de G que não está em T deve ser uma aresta de retorno, isto é, deve ligar um vértice a um de seus ancestrais ou a um de seus descendentes em T.

40. Seja G um grafo conexo. Mostre que, se T for uma árvore geradora de G construída usando busca em largura, então uma aresta de G que não está em T deve ligar vértices no mesmo nível ou em níveis que difiram em 1 na árvore geradora.

41. Para quais grafos a busca em profundidade e a busca em largura produzem árvores geradoras idênticas, não importando qual vértice é escolhido como raiz da árvore? Justifique sua resposta.

42. Use o Exercício 39 para demonstrar que, se G for um grafo simples conexo com n vértices e G não contiver um caminho simples de comprimento k, então ele contém, no máximo, $(k-1)n$ arestas.

43. Use indução matemática para demonstrar que a busca em largura visita os vértices em ordem de seus níveis na árvore geradora resultante.

44. Use pseudocódigo para descrever uma variação da busca em profundidade que associa o inteiro n ao n-ésimo vértice visitado na busca. Mostre que esta numeração corresponde à numeração dos vértices criada pelo percurso em pré-ordem da árvore geradora.

45. Use pseudocódigo para descrever uma variação da busca em largura que associa o inteiro m ao m-ésimo vértice visitado na busca.

***46.** Suponha que G seja um grafo orientado e T seja uma árvore geradora construída usando busca em largura. Mostre que toda aresta de G conecta dois vértices em um mesmo nível, um vértice a um vértice em um nível um a menos, ou um vértice a um vértice em algum nível maior.

47. Mostre que, se G for um grafo orientado e T for uma árvore geradora construída usando busca em profundidade, então toda aresta que não estiver na árvore geradora é uma **aresta progressiva** (*forward edge*) que conecta um ancestral a seu descendente, uma **aresta de retorno** (*back edge*) que conecta um descendente a um ancestral ou uma **aresta cruzada** (*cross edge*) que conecta um vértice a um vértice em uma subárvore previamente visitada.

***48.** Descreva uma variação da busca em profundidade que associa o menor inteiro positivo disponível a um vértice quando o algoritmo terminou completamente com aquele vértice. Mostre que, nesta enumeração, cada vértice tem um número maior do que seus filhos e que os filhos têm números crescentes da esquerda para a direita.

Sejam T_1 e T_2 árvores geradoras de um grafo. A **distância** entre T_1 e T_2 é o numero de arestas em T_1 e T_2 que não são comuns a T_1 e T_2.

49. Encontre a distância entre cada par de árvores geradoras mostradas nas Figuras 3(c) e 4 do grafo G mostrado na Figura 2.

***50.** Suponha que T_1, T_2 e T_3 sejam árvores geradoras de um grafo simples G. Mostre que a distância entre T_1 e T_3 não excede a soma da distância entre T_1 e T_2 e da distância entre T_2 e T_3.

****51.** Suponha que T_1 e T_2 sejam árvores geradoras de um grafo simples G. Além disso, suponha que e_1 seja uma aresta de T_1 que não esteja em T_2. Mostre que existe uma aresta e_2 em T_2 que não está em T_1, tal que T_1 continua sendo uma árvore geradora se e_1 for removida dela e e_2 for acrescentada a ela e T_2 continua sendo uma árvore geradora se e_2 for removida dela e e_1 for acrescentada a ela.

***52.** Mostre que é possível encontrar uma seqüência de árvores geradoras que leva de uma árvore geradora a qualquer outra por remoção de uma aresta e adição de outra, sucessivamente.

Uma **árvore geradora enraizada** de um grafo orientado é uma árvore enraizada que contém arestas do grafo tal que cada vértice do grafo é uma extremidade de uma das arestas da árvore.

53. Para cada um dos grafos orientados dos exercícios 18 a 23 da Seção 9.5 encontre uma árvore geradora enraizada do grafo ou determine que tal árvore não existe.

***54.** Mostre que um grafo orientado conexo, no qual cada vértice tem os mesmos grau de entrada e grau de saída, tem uma árvore geradora enraizada. [*Dica*: Use um ciclo euleriano.]

***55.** Dê um algoritmo para construir uma árvore geradora enraizada para grafos orientados conexos nos quais cada vértice tenha os mesmos grau de entrada e grau de saída.

***56.** Mostre que, se G for um grafo orientado e T for uma árvore geradora construída usando a busca em profundidade, então G contém um ciclo se e somente se G contiver uma aresta de retorno (veja o Exercício 47) relativa à árvore geradora T.

***57.** Use o Exercício 56 para construir um algoritmo que determine se um grafo orientado contém um ciclo.

10.5 Árvores Geradoras Mínimas

Introdução

Links

Uma companhia planeja construir uma rede de comunicações conectando seus cinco centros de computação. Cada par desses centros pode ser ligado por uma linha telefônica alugada. Quais ligações devem ser feitas para garantir que exista um caminho entre quaisquer dois centros de computação, de modo que o custo total da rede seja minimizado? Podemos modelar este problema usando o grafo com pesos mostrado na Figura 1, no qual os vértices representam os centros de computação, as arestas representam possíveis linhas alugadas e os pesos nas arestas são as taxas de aluguel mensais das linhas representadas pelas arestas. Podemos resolver este problema encontrando uma árvore geradora de modo que a soma dos pesos nas arestas da árvore seja minimizada. Tal árvore geradora é chamada de **árvore geradora mínima**.

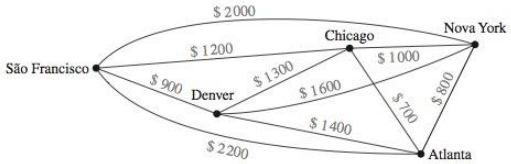


FIGURA 1 Um Grafo com Pesos que Mostra os Custos Mensais com Aluguel de Linhas para uma Rede de Computadores.

Algoritmos para Árvores Geradoras Mínimas

Uma grande variedade de problemas é resolvida encontrando uma árvore geradora em um grafo com pesos, de modo que a soma dos pesos das arestas da árvore seja mínima.

DEFINIÇÃO 1 Uma *árvore geradora mínima* em um grafo com pesos conexo é uma árvore geradora com a menor soma possível de pesos de suas arestas.

Demo

Apresentaremos dois algoritmos para a construção de árvores geradoras mínimas. Ambos prosseguem pela adição sucessiva de arestas de menor peso entre aquelas com uma propriedade especificada que ainda não tenha sido usada. Ambos são algoritmos vorazes. Lembre-se, da Seção 3.1, de que um algoritmo voraz é um procedimento que faz uma escolha ótima em cada um de seus passos. A otimização em cada passo não garante que a solução ótima geral seja produzida. Entretanto, os dois algoritmos apresentados nesta seção para a construção de árvores geradoras mínimas são algoritmos vorazes que, de fato, produzem soluções ótimas.

Links

O primeiro algoritmo que discutiremos foi dado por Robert Prim em 1957, embora as idéias básicas deste algoritmo tenham uma origem anterior. Para executar o **algoritmo de Prim**, comece escolhendo qualquer aresta com menor peso, colocando-a na árvore geradora. Sucessivamente, adicione à árvore arestas de peso mínimo que sejam incidentes a um vértice já na árvore e que não formem um ciclo simples com as arestas já na árvore. Pare quando tiverem sido adicionadas $n - 1$ arestas.

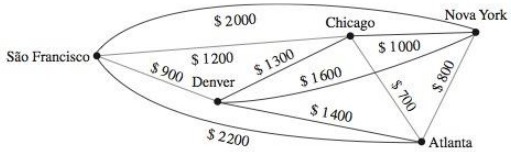
Mais tarde, nesta seção, demonstraremos que este algoritmo produz uma árvore geradora mínima para qualquer grafo conexo com pesos. O Algoritmo 1 dá uma descrição em pseudocódigo do algoritmo de Prim.

ALGORITMO 1 Algoritmo de Prim.

procedure *Prim*(G: grafo não orientado conexo com pesos com n vértices)
T := uma aresta com peso mínimo
for $i := 1$ **to** $n - 2$
begin
  e := uma aresta de peso mínimo incidente em um vértice em T e que não forme um ciclo simples em T se adicionada a T
  $T := T$ com e adicionada
end {T é uma árvore geradora mínima de G}

Observe que a escolha de uma aresta para adicionar em um estágio do algoritmo não está determinada quando houver mais de uma aresta com o mesmo peso que satisfaça os critérios apropriados. Precisamos ordenar as arestas para tornar a escolha determinística. Não nos preocuparemos com isso no restante da seção. Observe também que pode haver mais do que uma árvore gerado-

ROBERT CLAY PRIM (NASCIDO EM 1921) Robert Prim, nascido em Sweetwater, no Texas, se formou em engenharia elétrica em 1941 e obteve seu doutorado em matemática em 1949, na Universidade de Princeton. Ele foi engenheiro da General Eletric Company de 1941 a 1944, engenheiro e matemático no United States Naval Ordnance Lab de 1944 até 1949 e pesquisador associado na Universidade de Princeton de 1948 a 1949. Entre outros cargos que exerceu, estão: diretor de pesquisas mecânicas e matemáticas no Bell Telephone Laboratories de 1958 a 1961 e vice-presidente de pesquisa da Sandia Corporation. Atualmente, está aposentado.



Escolha	Aresta	Custo
1	{Chicago, Atlanta}	$ 700
2	{Atlanta, Nova York}	$ 800
3	{Chicago, São Francisco}	$ 1200
4	{São Francisco, Denver}	$ 900
	Total:	$ 3600

FIGURA 2 Uma Árvore Geradora Mínima para o Grafo com Pesos na Figura 1.

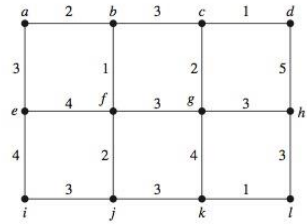


FIGURA 3 Um Grafo com Pesos.

ra mínima para um dado grafo simples com pesos conexo. (Veja o Exercício 9.) Os exemplos 1 e 2 ilustram como o algoritmo de Prim é usado.

EXEMPLO 1 Use o algoritmo de Prim para projetar uma rede de comunicação de custo mínimo que ligue todos os computadores representados pelo grafo na Figura 1.

Solução: Resolvemos este problema encontrando uma árvore geradora mínima no grafo da Figura 1. O algoritmo de Prim é executado escolhendo uma aresta inicial de peso mínimo e sucessivamente adicionando arestas de pesos mínimos que são incidentes a um vértice na árvore e que não formem um ciclo simples. As arestas em cinza na Figura 2 mostram uma árvore geradora mínima produzida pelo algoritmo de Prim, com a escolha feita em cada passo explicitada. ◀

EXEMPLO 2 Use o algoritmo de Prim para encontar uma árvore geradora mínima no grafo mostrado na Figura 3.

Solução: Uma árvore geradora mínima construída usando o algoritmo de Prim é mostrada na Figura 4. As arestas sucessivas escolhidas também estão mostradas. ◀

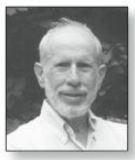
Links

O segundo algoritmo que discutiremos foi descoberto por Joseph Kruskal em 1956, embora as idéias básicas que ele use tenham sido descritas muito antes. Para executar o **algoritmo de Kruskal**, escolha uma aresta no grafo com peso mínimo.

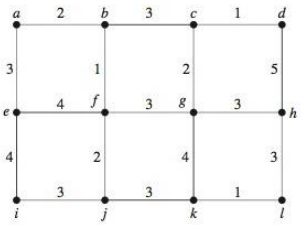
Adicione sucessivamente arestas com peso mínimo que não formem um ciclo simples com as arestas já escolhidas. Pare depois de $n - 1$ arestas terem sido selecionadas.

A demonstração de que o algoritmo de Kruskal produz uma árvore geradora mínima para todo grafo com pesos conexo é deixada como um exercício no final desta seção. O pseudocódigo para o algoritmo de Kruskal é dado no Algoritmo 2.

Links



JOSEPH BERNARD KRUSKAL (NASCIDO EM 1928) Joseph Kruskal, nascido na cidade de Nova York, freqüentou a Universidade de Chigago e obteve seu doutorado na Universidade de Princeton, em 1954. Ele foi instrutor de matemática em Princeton e na Universidade de Wisconsin, e mais tarde foi professor assistente na Universidade de Michigan. Em 1959, tornou-se membro da equipe técnica do Bell Laboratories, um cargo que ele ainda mantém. Seus interesses atuais em pesquisa incluem lingüística e psicometria estatísticas. Além de seu trabalho em árvores geradoras mínimas, Kruskal também é conhecido por suas contribuições para mudanças de escala multidimensionais. Kruskal descobriu seu algoritmo para produzir árvores geradoras mínimas quando estava em seu segundo ano de pós-graduação. Ele não tinha certeza de que seu trabalho de duas páginas e meia neste assunto deveria ser publicado, mas foi convencido por outros a publicá-lo.



(a)

Escolha	Aresta	Peso
1	{b, f}	1
2	{a, b}	2
3	{f, j}	2
4	{a, e}	3
5	{i, j}	3
6	{f, g}	3
7	{c, g}	2
8	{c, d}	1
9	{g, h}	3
10	{h, l}	3
11	{k, l}	1
	Total:	24

(b)

FIGURA 4 Uma Árvore Geradora Mínima Produzida Usando o Algoritmo de Prim.

ALGORITMO 2 Algoritmo de Kruskal.

procedure *Kruskal*(*G*: grafo não orientado conexo com pesos com *n* vértices)
$T :=$ grafo vazio
for $i := 1$ **to** $n - 1$
begin
  $e :=$ uma aresta qualquer em G de peso mínimo que não forme
    um ciclo simples quando adicionada a T
  $T := T$ com e adicionada
end {T é uma árvore geradora mínima de G}

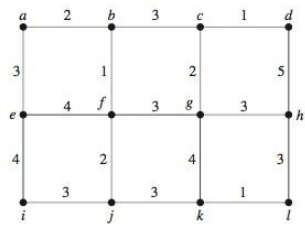
O leitor deve observar a diferença entre os algoritmos de Prim e de Kruskal. No algoritmo de Prim, arestas de peso mínimo que são incidentes a um vértice já na árvore e que não formem um ciclo são escolhidas; enquanto no algoritmo de Kruskal arestas de peso mínimo que não são necessariamente incidentes a um vértice já na árvore e que não formem um ciclo são escolhidas. Observe que no algoritmo de Prim, se as arestas não estiverem ordenadas, pode haver mais de uma escolha para a aresta a ser adicionada em um estágio deste procedimento. Conseqüentemente, as arestas precisam estar ordenadas para o procedimento ser determinístico. O Exemplo 3 ilustra como o algoritmo de Kruskal é usado.

EXEMPLO 3 Use o algoritmo de Kruskal para encontrar uma árvore geradora mínima no grafo com pesos mostrado na Figura 3.

Exemplos Extras

Solução: Uma árvore geradora mínima e as escolhas das arestas em cada estágio do algoritmo de Kruskal estão mostradas na Figura 5. ◀

OBSERVAÇÃO HISTÓRICA Joseph Kruskal e Robert Prim desenvolveram seus algoritmos para a construção de árvores geradoras mínimas em meados da década de 1950. Entretanto, eles não foram as primeiras pessoas a descobrir esses algoritmos. Por exemplo, o trabalho do antropologista Jan Czekanowski, em 1909, contém muitas das idéias necessárias para encontrar árvores geradoras mínimas,. Em 1926, Otakar Boruvka descreveu métodos para construir árvores geradoras mínimas em trabalho relacionado à construção de redes de energia elétrica.



(a)

Escolha	Aresta	Peso
1	{c, d}	1
2	{k, l}	1
3	{b, f}	1
4	{c, g}	2
5	{a, b}	2
6	{f, j}	2
7	{b, c}	3
8	{j, k}	3
9	{g, h}	3
10	{i, j}	3
11	{a, e}	3
	Total:	24

(b)

FIGURA 5 Uma Árvore Geradora Mínima Produzida pelo Algoritmo de Kruskal.

Demonstraremos agora que o algoritmo de Prim produz uma árvore geradora mínima de um grafo com pesos conexo.

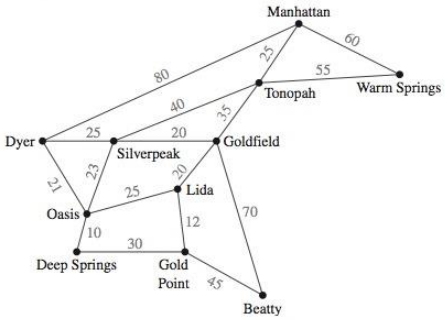
Demonstração: Seja G um grafo com pesos conexo. Suponha que as arestas sucessivas escolhidas pelo algoritmo de Prim são $e_1, e_2, \ldots, e_{n-1}$. Seja S a árvore com $e_1, e_2, \ldots, e_{n-1}$ como suas arestas, e seja S_k a árvore com $e_1, e_2, \ldots, e_k$ como suas arestas. Seja T uma árvore geradora mínima de G que contenha as arestas $e_1, e_2, \ldots, e_k$, em que k é o máximo inteiro com a propriedade que existe uma árvore geradora mínima que contém as primeiras k arestas escolhidas pelo algoritmo de Prim. O Teorema estará demonstrado se pudermos mostrar que $S = T$.

Suponha que $S \neq T$, de modo que $k < n - 1$. Conseqüentemente, T contém $e_1, e_2, \ldots, e_k$ mas não e_{k+1}. Considere o grafo formado por T junto com e_{k+1}. Como este grafo é conexo e tem n arestas, arestas demais para ser uma árvore, ele deve conter um ciclo simples. Este ciclo simples deve conter e_{k+1}, pois não existia nenhum ciclo simples em T. Além disso, deve existir uma aresta no ciclo simples que não pertence a S_{k+1}, pois S_{k+1} é uma árvore. Começando em uma extremidade de e_{k+1} que também seja uma extremidade de uma das arestas $e_1, \ldots, e_k$ e percorrendo o ciclo até atingirmos uma aresta que não esteja em S_{k+1}, podemos encontrar uma aresta e que não está em S_{k+1} que tem uma extremidade que também é uma das extremidades de $e_1, e_2, \ldots, e_k$. Removendo e de T e adicionando e_{k+1}, obtemos uma árvore T' com $n - 1$ arestas (é uma árvore, pois não tem ciclos simples). Observe que a árvore T' contém $e_1, e_2, \ldots, e_k, e_{k+1}$. Além disso, como e_{k+1} foi escolhido pelo algoritmo de Prim no $(k+1)$-ésimo passo e e também estava disponível naquele passo, o peso de e_{k+1} é menor que ou igual ao peso de e. Desta observação segue que T' também é uma árvore geradora mínima, pois a soma dos pesos de suas arestas não excede a soma dos pesos das arestas de T. Isso contradiz a escolha de k como o máximo inteiro tal que exista uma árvore geradora mínima que contenha $e_1, \ldots, e_k$. Portanto, $k = n - 1$ e $S = T$. Segue que o algoritmo de Prim produz uma árvore geradora mínima. ◁

Pode ser mostrado (veja [CoLeRiSt01]) que, para encontrar uma árvore geradora mínima de um grafo com e arestas e v vértices, o algoritmo de Kruskal pode ser executado usando $O(e \log e)$ operações e o algoritmo de Prim pode ser executado usando $O(e \log v)$ operações. Conseqüentemente, é preferível usar o algoritmo de Kruskal para grafos que são **esparsos**, isto é, nos quais e é muito pequeno quando comparado a $C(v, 2) = v(v-1)/2$, o número total de arestas possíveis em um grafo não orientado com v vértices. Caso contrário, há pouca diferença na complexidade destes dois algoritmos.

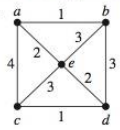
Exercícios

1. As estradas representadas por este grafo são todas não pavimentadas. Os comprimentos das estradas entre pares de cidades são representados por pesos nas arestas. Quais estradas devem ser pavimentadas de modo que haja um caminho de estradas pavimentadas entre cada par de cidades e que um comprimento mínimo de estrada seja pavimentado? (*Observação:* Estas cidades estão em Nevada.)

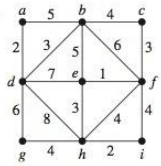


Nos exercícios 2 a 4, use o algoritmo de Prim para encontrar uma árvore geradora mínima para o grafo com pesos dado.

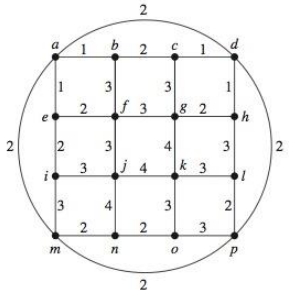
2.



3.



4.



5. Use o algoritmo de Kruskal para projetar a rede de comunicações descrita no começo da seção.

6. Use o algoritmo de Kruskal para encontrar uma árvore geradora mínima para o grafo com pesos do Exercício 2.

7. Use o algoritmo de Kruskal para encontrar uma árvore geradora mínima para o grafo com pesos do Exercício 3.

8. Use o algoritmo de Kruskal para encontrar uma árvore geradora mínima para o grafo com pesos do Exercício 4.

9. Encontre um grafo simples com pesos conexo com o número mínimo de arestas possível que tenha mais de uma árvore geradora mínima.

10. Uma **floresta geradora mínima** em um grafo com pesos é uma floresta geradora com peso mínimo. Explique como os algoritmos de Prim e de Kruskal podem ser adaptados para construir uma floresta geradora mínima.

Uma **árvore geradora máxima** de um grafo não orientado com pesos conexo é uma árvore geradora com o maior peso possível.

11. Desenvolva um algoritmo parecido com o algoritmo de Prim para construir uma árvore geradora máxima de um grafo com pesos conexo.

12. Desenvolva um algoritmo parecido com o algoritmo de Kruskal para construir uma árvore geradora máxima de um grafo com pesos conexo.

13. Encontre uma árvore geradora máxima para o grafo com pesos do Exercício 2.

14. Encontre uma árvore geradora máxima para o grafo com pesos do Exercício 3.

15. Encontre uma árvore geradora máxima para o grafo com pesos do Exercício 4.

16. Encontre a segunda rede de comunicações menos cara que ligue os cinco centros de computação do problema apresentado no início desta seção.

***17.** Desenvolva um algoritmo para encontrar a segunda menor árvore geradora em um grafo com pesos conexo.

***18.** Mostre que uma aresta com menor peso em um grafo com pesos conexo deve ser parte de qualquer árvore geradora mínima.

19. Mostre que existe uma única árvore geradora mínima em um grafo com pesos conexo se os pesos das arestas forem todos diferentes.

20. Suponha que a rede de computadores que liga as cidades na Figura 1 deve conter uma ligação direta entre Nova York e Denver. Que outras ligações devem ser incluídas, de modo que exista uma ligação entre cada dois centros de computadores e que o custo seja minimizado?

21. Encontre uma árvore geradora com peso total mínimo que contenha as arestas $\{e, i\}$ e $\{g, k\}$ no grafo com pesos da Figura 3.

22. Descreva um algoritmo para encontrar uma árvore geradora com peso mínimo que contenha um conjunto especificado de arestas em um grafo simples não orientado com pesos conexo.

23. Expresse o algoritmo desenvolvido no Exercício 22 em pseudocódigo.

O **algoritmo de Sollin** produz uma árvore geradora mínima de um grafo simples com pesos conexo $G = (V, E)$ adicionando sucessivamente grupos de arestas. Suponha que os vértices em V estejam ordenados. Isso produz uma ordenação das arestas em que $\{u_0, v_0\}$ precede $\{u_1, v_1\}$ se u_0 preceder u_1 ou se $u_0 = u_1$ e v_0 preceder v_1. O algoritmo começa escolhendo simultaneamente a aresta de menor peso incidente a cada vértice. A primeira aresta na ordenação é escolhida em caso de empate. Isso produz um grafo sem nenhum ciclo simples, ou seja, uma floresta de árvores (o Exercício 24 pede uma demonstração deste fato). A seguir, escolha simultaneamente para cada árvore da floresta a aresta de menor peso entre um vértice nesta árvore e um vértice em uma árvore diferente. Novamente a primeira aresta na ordenação é escolhida em caso de empate. (Isso produz um grafo sem nenhum ciclo simples que contém menos árvores do que havia antes deste passo; veja o Exercício 24.) Continue este processo de adicionar arestas conectando árvores até que tenham sido escolhidas $n - 1$ arestas. Neste estágio, terá sido construída uma árvore geradora mínima.

***24.** Mostre que a adição de arestas em cada estágio do algoritmo de Sollin produz uma floresta.

25. Use o algoritmo de Sollin para produzir uma árvore geradora mínima para o grafo com pesos mostrado na
a) Figura 1.
b) Figura 3.

***26.** Expresse o algoritmo de Sollin em pseudocódigo.

****27.** Demonstre que o algoritmo de Sollin produz uma árvore geradora mínima em um grafo com pesos não orientado conexo.

***28.** Mostre que o primeiro passo no algoritmo de Sollin produz uma floresta que contém, pelo menos, $\lceil n/2 \rceil$ arestas.

***29.** Mostre que, se existirem r árvores na floresta em algum estágio intermediário do algoritmo de Sollin, então, pelo menos, $\lceil r/2 \rceil$ arestas serão adicionadas na próxima iteração do algoritmo.

***30.** Mostre que restam não mais de $\lfloor n/2^k \rfloor$ árvores depois de o primeiro passo do algoritmo de Sollin ser executado e depois de o segundo passo do algoritmo ser executado $k - 1$ vezes.

***31.** Mostre que o algoritmo de Sollin exige no máximo $\log n$ iterações para produzir uma árvore geradora mínima de um grafo com pesos não orientado conexo com n vértices.

32. Demonstre que o algoritmo de Kruskal produz árvores geradoras mínimas.

Termos-chave e Resultados

TERMOS

árvore: um grafo não orientado conexo sem ciclos simples

floresta: um grafo não orientado sem ciclos simples

árvore enraizada: um grafo orientado com um vértice especificado, chamado de raiz, tal que existe um único caminho para qualquer outro vértice a partir dessa raiz

subárvore: um subgrafo de uma árvore que também é uma árvore

pai de v em uma árvore enraizada: o vértice u tal que (u, v) é uma aresta da árvore enraizada

filho de um vértice v em uma árvore com raiz: qualquer vértice com v como seu pai

irmão de um vértice v em uma árvore enraizada: um vértice com o mesmo pai que v

ancestral de um vértice v em uma árvore enraizada: qualquer vértice no caminho da raiz até v

descendente de um vértice v em uma árvore enraizada: qualquer vértice que tenha v como ancestral

vértice interno: um vértice que tem filhos

folha: um vértice sem filhos

nível de um vértice: o comprimento do caminho da raiz a este vértice

altura de uma árvore: o maior nível de todos os vértices da árvore

árvore de grau m: uma árvore com a propriedade de que todo vértice interno não tem mais de m filhos

árvore de grau m cheia: uma árvore com a propriedade de que todo vértice interno tem exatamente m filhos

árvore binária: uma árvore de grau m com $m = 2$ (cada filho pode ser designado como filho esquerdo ou direito de seu pai)

árvore ordenada: uma árvore na qual os filhos de cada vértice interno são linearmente ordenados

árvore balanceada: uma árvore na qual toda folha está no nível h ou $h - 1$, em que h é a altura da árvore

árvore de busca binária: uma árvore binária na qual os vértices estão rotulados com itens de modo que um rótulo de um vértice é maior que os rótulos de todos os vértices na subárvore esquerda deste vértice e é menor do que os rótulos de todos os vértices na subárvore direita deste vértice

árvore de decisão: uma árvore enraizada em que cada vértice representa uma possível saída de uma decisão e as folhas representam as soluções possíveis

código de prefixo: um código que tem a propriedade de que o código de um caractere nunca é um prefixo do código de um outro caractere

estratégia minimax: a estratégia em que o primeiro jogador e o segundo jogador se movem para posições representadas por um filho com valor máximo e mínimo, respectivamente

valor de um vértice em uma árvore de jogo: para uma folha, o prêmio do primeiro jogador quando o jogo termina na posição representada por esta folha; para um vértice interno, o máximo ou o mínimo dos valores de seus filhos, para um vértice interno em um nível par ou ímpar, respectivamente

percurso na árvore: uma listagem de todos os vértices de uma árvore

percurso em pré-ordem: uma listagem de todos os vértices de uma árvore enraizada ordenada definida recursivamente – a raiz é listada, seguida pela primeira subárvore, seguida pelas outras subárvores na ordem em que elas ocorrem da esquerda para a direita

percurso em ordem simétrica: uma listagem de todos os vértices de uma árvore enraizada ordenada definida recursivamente – a primeira subárvore é listada, seguida pela raiz, seguida pelas outras subárvores na ordem em que elas ocorrem da esquerda para a direita

percurso em pós-ordem: uma listagem de todos os vértices de uma árvore enraizada ordenada definida recursivamente – as subárvores são listadas na ordem em que elas ocorrem da esquerda para a direita, seguidas pela raiz

notação infixa: a forma de uma expressão (incluindo um conjunto completo de parênteses) obtida a partir de um percurso em ordem simétrica da árvore binária que representa esta expressão

notação prefixa (ou polonesa): a forma de uma expressão obtida a partir de um percurso em pré-ordem da árvore que representa esta expressão

notação pós-fixa (ou polonesa reversa): a forma de uma expressão obtida a partir de um percurso em pós-ordem da árvore que representa esta expressão

árvore geradora: uma árvore que contém todos os vértices de um grafo

árvore geradora mínima: uma árvore geradora com a menor soma possível dos pesos de suas arestas

algoritmo voraz: um algoritmo que otimiza fazendo a escolha ótima em cada passo

RESULTADOS

Um grafo é uma árvore se e somente se existir um único caminho simples entre quaisquer de seus vértices.

Uma árvore com n vértices tem $n - 1$ arestas.

Uma árvore de grau m cheia com i vértices internos tem $mi + 1$ vértices.

A relação entre o número de vértices, folhas e vértices internos de uma árvore de grau m cheia (veja o Teorema 4 na Seção 10.1).

Existem, no máximo, m^h folhas em uma árvore de grau m de altura h.

Se uma árvore de grau m tiver l folhas, sua altura h será, pelo menos, $\lceil \log_m l \rceil$. Se a árvore também for cheia e balanceada, então sua altura será $\lceil \log_m l \rceil$.

codificação de Huffman: um procedimento para construir um código binário ótimo para um conjunto de símbolos, dadas as freqüências desses símbolos

busca em profundidade ou *backtracking*: um procedimento para construir uma árvore geradora que adiciona arestas para formar um caminho até que isso não seja mais possível, e então retroceder no caminho até que seja encontrado um vértice no qual um novo caminho possa ser formado

busca em largura: um procedimento para construir uma árvore geradora que adiciona sucessivamente todas as arestas incidentes ao último conjunto de arestas adicionadas, a menos que um ciclo simples seja formado

algoritmo de Prim: um procedimento para produzir uma árvore geradora mínima em um grafo com pesos que adiciona sucessivamente arestas com pesos mínimos entre todas as arestas incidentes a um vértice já na árvore tal que nenhuma aresta produza um ciclo simples quando adicionada

algoritmo de Kruskal: um procedimento para produzir uma árvore geradora mínima em um grafo com pesos que adiciona sucessivamente arestas com pesos mínimos que ainda não estejam na árvore tal que nenhuma aresta produza um ciclo simples quando adicionada

Questões de Revisão

1. a) Defina uma árvore. **b)** Defina uma floresta.

2. Podem existir dois caminhos simples diferentes entre os vértices de uma árvore?

3. Dê, pelo menos, três exemplos de como as árvores são usadas na modelagem.

4. a) Defina uma árvore enraizada e a raiz dessa árvore.
b) Defina o pai de um vértice e um filho de um vértice em uma árvore enraizada.
c) O que são um vértice interno, uma folha e uma subárvore em uma árvore enraizada?
d) Desenhe uma árvore enraizada com, pelo menos, dez vértices, na qual o grau de cada vértice não exceda três. Identifique a raiz, o pai de cada vértice, os filhos de cada vértice, os vértices internos e as folhas.

5. a) Quantas arestas tem uma árvore com n vértices?
b) O que você precisa saber para determinar o número de arestas em uma floresta com n vértices?

6. a) Defina uma árvore de grau m cheia.
b) Quantos vértices tem uma árvore de grau m cheia se ela tiver i vértices internos? Quantas folhas a árvore tem?

7. a) O que é a altura de uma árvore enraizada?
b) O que é uma árvore balanceada?
c) Quantas folhas pode ter uma árvore de grau m de altura h?

8. a) O que é uma árvore de busca binária?
b) Descreva um algoritmo para construir uma árvore de busca binária.
c) Forme uma árvore de busca binária para as palavras *vireo, warbler, egret, grosbeak, nuthatch* e *kingfisher*.

9. a) O que é um código de prefixo?
b) Como um código de prefixo pode ser representado por uma árvore binária?

10. a) Defina um percurso em árvore em pré-ordem, em ordem simétrica e em pós-ordem.
b) Dê um exemplo de percurso em pré-ordem, em pós-ordem e em ordem simétrica de uma árvore binária de sua escolha com, pelo menos, 12 vértices.

11. a) Explique como usar percursos em pré-ordem, em ordem simétrica e em pós-ordem para encontrar as formas prefixa, infixa e pós-fixa de uma expressão aritmética.
b) Desenhe a árvore enraizada ordenada que represente $((x - 3) + ((x/4) + (x - y)) \uparrow 3))$.
c) Encontre as formas prefixa e pós-fixa da expressão da parte (b).

12. Mostre que o número de comparações usadas por um algoritmo de ordenação é, pelo menos, $\lceil \log n! \rceil$.

13. a) Descreva o algoritmo de codificação de Huffman para construir um código ótimo para um conjunto de símbolos, dadas as freqüências desses símbolos.
b) Use codificação de Huffman para encontrar um código ótimo para estes símbolos e freqüências: A: 0, 2, B: 0,1, C: 0,3, D: 0,4.

14. Desenhe a árvore de jogo para o nim se a posição inicial consiste em duas pilhas com uma e quatro pedras, respectivamente. Quem ganha o jogo se ambos os jogadores seguirem a estratégia ótima?

15. a) O que é uma árvore geradora de um grafo simples?

b) Quais grafos simples têm árvores geradoras?

c) Descreva, pelo menos, duas aplicações diferentes que exigem que seja encontrada uma árvore geradora de um grafo simples.

16. a) Descreva dois algoritmos diferentes para encontrar uma árvore geradora em um grafo simples.

b) Ilustre como os dois algoritmos que você descreveu em (a) podem ser usados para encontrar a árvore geradora de um grafo simples, usando um grafo de sua escolha com, pelo menos, oito vértices e quinze arestas.

17. a) Explique como o *backtracking* pode ser usado para determinar se um grafo simples pode ser colorido usando n cores.

b) Mostre, com um exemplo, como o *backtracking* pode ser usado para mostrar que um grafo com um número cromático igual a 4 não pode ser colorido usando três cores, mas pode ser colorido usando quatro cores.

18. a) O que é a árvore geradora mínima de um grafo com pesos conexo?

b) Descreva, pelo menos, duas aplicações diferentes que exijam que seja encontrada uma árvore geradora mínima de um grafo com pesos conexo.

19. a) Descreva o algoritmo de Kruskal e o algoritmo de Prim para encontrar árvores geradoras mínimas.

b) Ilustre como o algoritmo de Kruskal e o algoritmo de Prim são usados para encontrar uma árvore geradora mínima, usando um grafo com pesos com, pelo menos, oito vértices e quinze arestas.

Exercícios Complementares

***1.** Mostre que um grafo simples é uma árvore se e somente se ele não contiver nenhum ciclo simples e a adição de uma aresta que liga dois vértices não adjacentes produzir um novo grafo que tem exatamente um ciclo simples (no qual ciclos que contêm as mesmas arestas não são considerados diferentes).

***2.** Quantas árvores enraizadas não isomorfas existem com seis vértices?

3. Mostre que toda árvore com, pelo menos, uma aresta deve ter, pelo menos, dois vértices pendentes.

4. Mostre que uma árvore com n vértices que tem $n - 1$ vértices pendentes deve ser isomorfa a $K_{1,n-1}$.

5. Qual é a soma dos graus dos vértices de uma árvore com n vértices?

***6.** Suponha que $d_1, d_2, ..., d_n$ sejam n inteiros positivos com soma $2n - 2$. Mostre que existe uma árvore que tem n vértices tal que o grau desses vértices sejam $d_1, d_2, ..., d_n$.

7. Mostre que toda árvore é um grafo planar.

8. Mostre que toda árvore é bipartida.

9. Mostre que toda floresta pode ser colorida usando duas cores.

Uma **B-árvore de grau *k*** é um árvore enraizada tal que todas as suas folhas estão em um mesmo nível, sua raiz tem no mínimo dois e no máximo k filhos, a menos que seja uma folha, e todo vértice interno diferente da raiz tem pelo menos $\lceil k/2 \rceil$, mas não mais do que k, filhos. Arquivos de computador podem ser acessados eficientemente quando B-árvores são usadas para representá-los.

10. Desenhe três B-árvores de grau 3 diferentes com altura 4.

***11.** Dê um limitante superior e um limitante inferior para o número de folhas em uma B-árvore de grau k com altura h.

***12.** Dê um limitante superior e um limitante inferior para a altura de uma B-árvore de grau k com n folhas.

As **árvores binomiais** B_i, $i = 0, 1, 2, ...$, são árvores enraizadas ordenadas definidas recursivamente:

Passo-base: A árvore binomial B_0 é a árvore com um único vértice.

Passo recursivo: Seja k um inteiro não negativo. Para construir a árvore binomial B_{k+1}, adicione uma cópia de B_k a uma segunda cópia de B_k adicionando uma aresta que torna a raiz da primeira cópia de B_k o filho mais à esquerda da segunda cópia de B_k.

13. Desenhe B_k para $k = 0, 1, 2, 3, 4$.

14. Quantos vértices B_k tem? Demonstre que sua resposta está correta.

15. Encontre a altura de B_k. Demonstre que sua resposta está correta.

16. Quantos vértices existem em B_k na profundidade j, em que $0 \leq j \leq k$? Justifique sua resposta.

17. Qual é o grau da raiz de B_k? Demonstre que sua resposta está correta.

18. Mostre que o vértice de maior grau em B_k é a raiz.

Uma árvore T com raiz é dita uma ***S_k*-árvore** se satisfizer esta definição recursiva. É uma S_0-árvore se tiver um vértice. Para $k > 0$, T é uma S_k-árvore se puder ser construída a partir de duas S_{k-1}-árvores, fazendo da raiz de uma a raiz da S_k-árvore e fazendo da raiz da outra o filho da raiz da primeira S_k-árvore.

19. Desenhe uma S_k-árvore para $k = 0, 1, 2, 3, 4$.

20. Mostre que uma S_k-árvore tem 2^k vértices e um único vértice no nível k. Este vértice no nível k é chamado de **alça** (*handle*, no inglês).

***21.** Suponha que T seja uma S_k-árvore com alça v. Mostre que T pode ser obtida de árvores disjuntas $T_0, T_1, ..., T_{k-1}$, em que v não está em nenhuma dessas árvores e em que T_i é uma S_i-árvore para $i = 0, 1, ..., k - 1$, conectando v a r_0 e r_i a r_{i+1} para $i = 0, 1, ..., k - 2$.

A listagem de vértices de uma árvore enraizada ordenada em **ordem de nível** começa com a raiz, seguida pelos vértices de ordem 1, da esquerda para a direita, pelos vértices de nível 2, da esquerda para a direita, e assim por diante.

22. Liste os vértices das árvores enraizadas ordenadas das figuras 3 e 9 da Seção 10.3 em ordem de nível.

23. Desenvolva um algoritmo para listar os vértices de uma árvore enraizada ordenada em ordem de nível.

***24.** Desenvolva um algoritmo para determinar se um conjunto de endereços globais podem ser os endereços das folhas de uma árvore enraizada.

25. Desenvolva um algoritmo para construir uma árvore enraizada a partir do endereço global de suas folhas.

Um **conjunto de corte** de um grafo é um conjunto de arestas tal que a remoção destas arestas produz um subgrafo com mais componentes conexas do que o grafo original, mas nenhum subconjunto próprio deste conjunto de arestas tem esta propriedade.

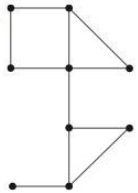
26. Mostre que um conjunto de corte de um grafo deve ter, pelo menos, uma aresta em comum com qualquer árvore geradora deste grafo.

Um **cactus** é um grafo conexo no qual nenhuma aresta está em mais de um ciclo simples que não passe por qualquer vértice diferente de seu vértice inicial mais do que uma vez ou seu vértice inicial diferente de seu vértice final (em que dois ciclos que contêm as mesmas arestas não são considerados diferentes.)

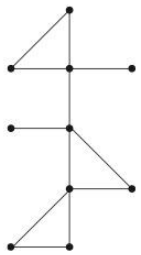
27. Quais destes grafos são *cactus*?

a)

b)



c)



28. Uma árvore é necessariamente um *cactus*?

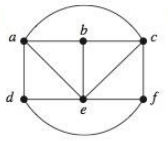
29. Mostre que um *cactus* é formado se adicionarmos um ciclo com novas arestas começando e terminando em um vértice de uma árvore.

***30.** Mostre que, se todo ciclo que não passe por qualquer vértice diferente de seu vértice inicial mais de uma vez em um grafo conexo contiver um número ímpar de arestas, então esse grafo deve ser um cactus.

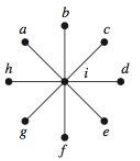
Uma **árvore geradora de grau restrito** de um grafo simples G é uma árvore geradora com a propriedade de que o grau de um vértice nesta árvore não pode exceder a algum limitante especificado. Árvores geradoras de grau restrito são úteis em modelos de sistemas de transporte, nos quais o número de estradas em uma intersecção é limitado, modelos de redes de computação, nos quais o número de links entrando em um nó é limitado, e assim por diante.

Nos exercícios 31 a 33, encontre uma árvore geradora de grau restrito do grafo dado, em que cada vértice tem grau menor que ou igual a 3, ou mostre que tal árvore geradora não existe.

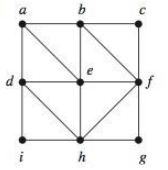
31.



32.



33.



34. Mostre que uma árvore geradora de grau restrito de um grafo simples, no qual cada vértice tem grau que não exceda a 2, consiste em um único caminho hamiltoniano no grafo.

35. Uma árvore com n vértices é dita **graciosa** se seus vértices puderem ser rotulados com os inteiros 1, 2, ..., n, de modo que os valores absolutos da diferença de rótulos dos vértices adjacentes são todos diferentes. Mostre que estas árvores são graciosas.

a)

b)

c)

d)

Uma **centopéia** é uma árvore que contém um caminho simples tal que cada vértice que não está contido nesse caminho é adjacente a um vértice neste caminho.

36. Quais dos grafos do Exercício 35 são centopéias?

37. Quantas centopéias não isomorfas com seis vértices existem?

****38.** **a)** Demonstre ou dê contra-exemplo de que todas as árvores cujas arestas formam um único caminho são graciosas.

b) Demonstre ou dê contra-exemplo de que todas as centopéias são graciosas.

39. Suponha que em uma seqüência longa de bits, a freqüência de ocorrência de um bit 0 é 0,9 e a freqüência de um bit 1 é 0,1 e os bits ocorrem independentemente.
- **a)** Construa um código de Huffman para os quatro blocos de dois bits, 00, 01, 10 e 11. Qual é o número médio de bits necessário para codificar uma seqüência de bits usando este código?
- **b)** Construa um código de Huffman para os oito blocos de três bits. Qual é o número médio de bits necessário para codificar uma sequência de bits usando este código?

40. Suponha que G seja um grafo orientado sem nenhum ciclo. Descreva como a busca em profundidade pode ser usada para fazer uma ordenação topológica dos vértices de G.

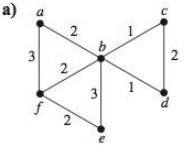
***41.** Suponha que e seja uma aresta em um grafo com pesos que é incidente em um vértice v tal que o peso de e não exceda o peso de qualquer outra aresta incidente em v. Mostre que existe uma árvore geradora mínima com esta aresta.

42. Três casais chegam ao banco de um rio. Cada uma das esposas é ciumenta e não confia em seu marido quando ele está com uma das outras esposas (e talvez com outras pessoas), mas não com ela. Como podem seis pessoas cruzar para o outro lado do rio usando um barco que transporta não mais de duas pessoas, de modo que nenhum marido fique sozinho com uma mulher que não seja a sua esposa? Use um modelo de teoria dos grafos.

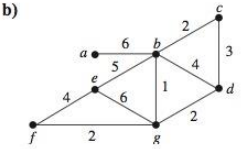
***43.** Mostre que, se nenhum par de arestas em um grafo com pesos tiver o mesmo peso, então a aresta com menor peso incidente no vértice v estará incluída em toda árvore geradora mínima.

44. Encontre uma árvore geradora mínima de cada um destes grafos em que o grau de cada vértice na árvore geradora não exceda a 2.

a)



b)



Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dada a matriz de adjacência de um grafo simples não orientado, determine se o grafo é uma árvore.

2. Dada a matriz de adjacência de uma árvore enraizada e um vértice na árvore, encontre pai, filhos, ancestrais, descendentes e nível desse vértice.

3. Dada a lista das arestas de uma árvore enraizada e um vértice na árvore, encontre pai, filhos, ancestrais, descendentes e nível desse vértice.

4. Dada uma lista de itens, construa uma árvore de busca binária que contenha estes itens.

5. Dada uma árvore de busca binária e um item, localize ou adicione este item na árvore de busca binária.

6. Dada uma lista ordenada de arestas de uma árvore enraizada ordenada, encontre os endereços globais de seus vértices.

7. Dada uma lista ordenada de arestas de uma árvore enraizada ordenada, liste seus vértices em pré-ordem, em ordem simétrica e em pós-ordem.

8. Dada uma expressão aritmética na forma prefixa, encontre o seu valor.

9. Dada uma expressão aritmética na forma pós-fixa, encontre o seu valor.

10. Dada a freqüência de símbolos, use a codificação de Huffman para encontrar um código ótimo para esses símbolos.

11. Dada uma posição inicial no jogo do nim, determine uma estratégia ótima para o primeiro jogador.

12. Dada a matriz de adjacência de um grafo simples não orientado conexo, encontre uma árvore geradora para esse grafo usando busca em profundidade.

13. Dada a matriz de adjacência de um grafo simples não orientado conexo, encontre uma árvore geradora para esse grafo usando busca em largura.

14. Dado um conjunto de inteiros positivos e um inteiro positivo N, use *backtracking* para encontrar um subconjunto desses inteiros que tem N como sua soma.

***15.** Dada a matriz de adjacência de um grafo simples não orientado, use *backtracking* para colorir o grafo com três cores, se isso for possível.

***16.** Dado um inteiro positivo n, resolva o problema das n-rainhas usando *backtracking*.

17. Dada a lista das arestas e seus pesos de um grafo conexo não orientado com pesos, use o algoritmo de Prim para encontrar uma árvore geradora mínima desse grafo.

18. Dada uma lista de arestas e seus pesos de um grafo conexo não orientado com pesos, use o algoritmo de Kruskal para encontrar uma árvore geradora mínima desse grafo.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Exiba todas as árvores com seis vértices.
2. Exiba um conjunto completo de árvores não isomorfas com sete vértices.
*3. Construa um código de Huffman para os símbolos com códigos ASCII dadas as freqüências de suas ocorrências em entradas representativas.
4. Calcule o número de árvores geradoras diferentes de K_n para $n = 1, 2, 3, 4, 5, 6$. Faça uma conjectura sobre uma fórmula para o número de tais árvores geradoras quando n for um número inteiro positivo.
5. Compare o número de comparações necessárias para ordenar listas de n elementos para $n = 100$, 1 000 e 10 000, em que os elementos são inteiros positivos escolhidos aleatoriamente, usando ordenação por seleção, ordenação por inserção, merge sort e ordenação rápida.
6. Calcule o número de maneiras diferentes que n rainhas podem ser arrumadas em um tabuleiro $n \times n$ de modo que nenhum par de rainhas possa atacar uma a outra para todos os inteiros positivos n que não excedam a 10.
*7. Encontre uma árvore geradora mínima do grafo que conecta as capitais dos 50 estados dos Estados Unidos entre si, em que o peso de cada aresta é a distância entre as cidades.
8. Desenhe a árvore de jogo completa para um jogo de damas em um tabuleiro 4×4.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Explique como Cayley usou árvores para enumerar o número de certos tipos de hidrocarbonetos.
2. Defina *árvores AVL* (às vezes conhecidas por *árvores balanceadas por altura*). Descreva como e por que as árvores AVL são usadas em uma variedade de algoritmos diferentes.
3. Defina *árvores quad* e explique como imagens podem ser representadas usando-as. Descreva como podemos girar imagens, fazer mudança de escala e transladá-las manipulando a árvore quad correspondente.
4. Defina um *heap* e explique como as árvores podem ser transformadas em *heaps*. Por que os *heaps* são úteis em ordenação?
5. Descreva algoritmos dinâmicos para compressão de dados baseados nas freqüências das letras à medida que elas mudam conforme os caracteres são sucessivamente lidos, como a codificação de Huffman adaptativa.
6. Explique como a *poda alpha-beta* pode ser usada para simplificar os cálculos do valor de uma árvore de jogo.
7. Descreva as técnicas usadas pelos programas de jogo de xadrez, como o Deep Blue.
8. Descreva o tipo de grafo conhecido como *mesh de árvores*. Explique como este grafo é usado em aplicações a sistemas muito grandes de integração e computação paralela.
9. Discuta os algoritmos usados em distribuição múltipla em IP para evitar laços entre os routers.
10. Descreva um algoritmo baseado em busca em profundidade para encontrar os pontos de articulação de um grafo.
11. Descreva um algoritmo baseado em busca em profundidade para encontrar componentes fortemente conexas de um grafo orientado.
12. Descreva as técnicas de busca usadas por crawlers e spiders em instrumentos de buscas diferentes na Web.
13. Descreva um algoritmo para encontrar a árvore geradora mínima de um grafo tal que o grau máximo de qualquer vértice na árvore geradora não exceda uma constante fixa k.
14. Compare e contraste alguns dos algoritmos de ordenação mais importantes em termos de sua complexidade e de quando são usados.
15. Discuta a história e as origens dos algoritmos para a construção de árvores geradoras mínimas.
16. Descreva algoritmos para a produção de árvores aleatórias.

CAPÍTULO

11 Álgebra Booleana

11.1 Funções Booleanas
11.2 Representação de Funções Booleanas
11.3 Portas Lógicas
11.4 Minimização de Circuitos

Os circuitos nos computadores e em outros aparelhos eletrônicos têm entradas, cada uma das quais ou é 0 ou 1 e produz saídas que também são 0s e 1s. Os circuitos podem ser construídos usando qualquer elemento básico que tenha dois estados diferentes. Esses elementos incluem chaves que podem estar na posição aberta ou fechada e aparelhos óticos que podem estar acesos ou não. Em 1938, Claude Shannon mostrou como as regras básicas da lógica, enunciadas pela primeira vez por George Boole em 1854, em seu *The Laws of Thought*, poderiam ser usadas para projetar circuitos. Essas regras formam a base da álgebra booleana. Neste capítulo, desenvolvemos as propriedades básicas dessa álgebra. A operação de um circuito é definida por uma função booleana que especifica o valor de uma saída para cada conjunto de entradas. O primeiro passo na construção de um circuito é representar a função booleana por uma expressão construída usando as operações básicas da álgebra booleana. Forneceremos um algoritmo para produzir essas expressões. A expressão que obtemos pode conter muito mais operações do que seriam necessárias para representar a função. Mais adiante, neste capítulo, veremos métodos para encontrar uma expressão com o número mínimo de somas e produtos que representam uma função booleana. Os procedimentos que desenvolveremos, mapas de Karnaugh e o método de Quine–McCluskey, são importantes no projeto de circuitos eficientes.

11.1 Funções Booleanas

Introdução

A álgebra booleana fornece as operações e as regras para trabalhar com o conjunto $\{0, 1\}$. Chaves eletrônicas e óticas podem ser estudadas usando esse conjunto e as regras da álgebra booleana. As três operações na álgebra booleana que mais usaremos são a complementação, a soma booleana e o produto booleano. O **complemento** de um elemento, indicado por uma barra, é definido por $\overline{0} = 1$ e $\overline{1} = 0$. A soma booleana, indicada por $+$ ou por *OU*, tem os seguintes valores:

$$1 + 1 = 1, \qquad 1 + 0 = 1, \qquad 0 + 1 = 1, \qquad 0 + 0 = 0.$$

O produto booleano, indicado por $\cdot$ ou por *E*, tem os seguintes valores:

$$1 \cdot 1 = 1, \qquad 1 \cdot 0 = 0, \qquad 0 \cdot 1 = 0, \qquad 0 \cdot 0 = 0.$$

Quando não houver perigo de confusão, o símbolo $\cdot$ pode ser omitido, do mesmo modo que escrevemos um produto algébrico. A menos que sejam usados parênteses, as regras de precedência para as operações booleanas são: primeiro, são calculados todos os complementos, seguidos pelos produtos booleanos, seguidos por todas as somas booleanas. Isso está ilustrado no Exemplo 1.

EXEMPLO 1 Encontre o valor de $1 \cdot 0 + \overline{(0 + 1)}$.

Solução: Usando as definições de complementação, de soma booleana e de produto booleano, segue que

$$\begin{aligned} 1 \cdot 0 + \overline{(0 + 1)} &= 0 + \overline{1} \\ &= 0 + 0 \\ &= 0. \end{aligned}$$

◀

O complemento, a soma boolena e o produto booleano correspondem aos operadores lógicos $\neg$, $\vee$ e $\wedge$, respectivamente, em que 0 corresponde a **F** (falso) e 1 corresponde a **V** (verdade). As igualdades na álgebra booleana podem ser traduzidas diretamente por equivalências de proposições compostas. Reciprocamente, equivalências de proposições compostas podem ser traduzidas por igualdades na álgebra booleana. Veremos mais tarde, nesta seção, por que essas traduções fornecem equivalências lógicas válidas e identidades na álgebra booleana. O Exemplo 2 ilustra a tradução da álgebra booleana para a lógica proposicional.

EXEMPLO 2 Traduza $1 \cdot 0 + \overline{(0 + 1)} = 0$, a igualdade encontrada no Exemplo 1, em uma equivalência lógica.

Solução: Obtemos uma equivalência lógica quando traduzimos cada 1 por um **V** e cada 0 por um **F**, cada soma booleana por uma disjunção, cada produto booleano por uma conjunção e cada complementação por uma negação. Obtemos

$$(\mathbf{V} \wedge \mathbf{F}) \vee \neg(\mathbf{V} \vee \mathbf{F}) \equiv \mathbf{F}.$$ ◀

O Exemplo 3 ilustra a tradução da lógica proposicional para a álgebra booleana.

EXEMPLO 3 Traduza a equivalência lógica $(\mathbf{V} \wedge \mathbf{V}) \vee \neg\mathbf{F} \equiv \mathbf{T}$ para uma identidade na álgebra booleana.

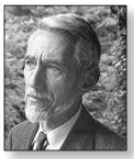
Solução: Obtemos uma identidade na álgebra booleana quando traduzimos cada **V** por 1, cada **F** por 0, cada disjunção por uma soma booleana, cada conjunção por um produto booleano e cada negação por uma complementação. Obtemos

$$(1 \cdot 1) + \overline{0} = 1.$$ ◀

Expressões Booleanas e Funções Booleanas

Seja $B = \{0, 1\}$. Então $B^n = \{(x_1, x_2, \ldots, x_n) \mid x_i \in B \text{ para } 1 \leq i \leq n\}$ é o conjunto de todas as n-uplas de 0s e 1s. A variável x é chamada de **variável booleana** se ela assume valores apenas em B, ou seja, se seus únicos valores possíveis são 0 e 1. Uma função de B^n para B é chamada de **função booleana de n variáveis**.

Links



CLAUDE ELWOOD SHANNON (1916–2001) Claude Shannon nasceu em Petoskey, Michigan, e cresceu em Gaylord, Michigan. Seu pai era negociante e juiz de sucessão e sua mãe era professora de línguas e diretora de escola de ensino médio. Shannon freqüentou a Universidade de Michigan, tendo se formado em 1936. Ele continuou seus estudos no M.I.T., onde obteve o trabalho de manutenção do analisador diferencial, um aparelho mecânico de computação que consistia em bastões e engrenagens, construído por seu orientador, Vannevar Bush. A dissertação de mestrado de Shannon, escrita em 1936, estudava os aspectos lógicos do analisador diferencial. Essa tese de mestrado apresentava a primeira aplicação da álgebra booleana ao projeto de circuitos de chaves; ela talvez seja a dissertação de mestrado mais famosa do século XX. Ele recebeu seu título de doutor do M.I.T. em 1940. Shannon ingressou no Bell Laboratories em 1940, onde trabalhou na transmissão eficiente de dados. Ele foi uma das primeiras pessoas a usar bits para representar informação. No Bell Laboratories, ele trabalhou na determinação da quantidade de tráfico que as linhas telefônicas podem suportar. Shannon fez muitas contribuições fundamentais à teoria da informação. No início da década de 1950, ele foi um dos fundadores do estudo da inteligência artificial. Ele ingressou no quadro de professores do M.I.T. em 1956, onde continuou seu estudo da teoria da informação.

Shannon tinha um lado não convencional. A invenção do Frisbee movido a foguete é atribuída a ele. Ele também é famoso por andar em um monociclo pelos corredores do Bell Laboratories ao mesmo tempo em que fazia malabarismos com quatro bolas. Shannon se aposentou quando tinha 50 anos e publicou artigos esporadicamente durante a década seguinte. Mais tarde, se concentrou em alguns projetos pessoais, tal como a construção de um pula-pula motorizado. Uma citação interessante de Shannon, publicada em *Omni Magazine* em 1987, é "Eu visualizo uma época na qual seremos para os robôs o que os cachorros são para os humanos. E estou torcendo pelas máquinas".

EXEMPLO 4 A função $F(x, y) = x\bar{y}$ do conjunto de pares ordenados de variáveis booleanas para o conjunto $\{0, 1\}$ é uma função booleana de duas variáveis com $F(1, 1) = 0$, $F(1, 0) = 1$, $F(0, 1) = 0$ e $F(0, 0) = 0$. Mostramos esses valores de F na Tabela 1. ◀

TABELA 1

x	y	$F(x, y)$
1	1	0
1	0	1
0	1	0
0	0	0

As funções booleanas podem ser representadas usando expressões formadas por variáveis e operações booleanas. As **expressões booleanas** nas variáveis $x_1, x_2, \ldots, x_n$ são definidas recursivamente como

$0, 1, x_1, x_2, \ldots, x_n$ são expressões booleanas;

se E_1 e E_2 são expressões booleanas, então $\overline{E}_1$, (E_1E_2) e $(E_1 + E_2)$ são expressões booleanas.

Cada expressão booleana representa uma função booleana. Os valores dessa função são obtidos substituindo as variáveis na expressão por 0 e 1. Na Seção 11.2, mostraremos que toda função booleana pode ser representada por uma expressão booleana.

EXEMPLO 5 Encontre os valores da função booleana representada por $F(x, y, z) = xy + \bar{z}$.

Solução: Os valores dessa função estão mostrados na Tabela 2. ◀

Observe que podemos representar uma função booleana graficamente distinguindo os vértices do n-cubo que corresponde às n-uplas de bits nos quais a função tem valor 1.

EXEMPLO 6 A função $F(x, y, z) = xy + \bar{z}$ de B^3 para B do Exemplo 3 pode ser representada distinguindo os vértices que correspondem às cinco 3-uplas (1, 1, 1), (1, 1, 0), (0, 0, 0), (0, 1, 0) e (0, 0, 0), nas quais $F(x, y, z) = 1$, como mostrado na Figura 1. Esses vértices estão mostrados pelos círculos pretos cheios. ◀

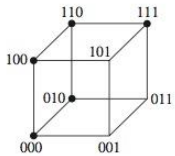


FIGURA 1

As funções booleanas F e G de n variáveis são iguais se e somente se $F(b_1, b_2, \ldots, b_n) = G(b_1, b_2, \ldots, b_n)$ sempre que $b_1, b_2, \ldots, b_n$ pertença a B. Duas expressões booleanas diferentes que representam a mesma função são ditas **equivalentes**. Por exemplo, as expressões booleanas xy, $xy + 0$ e $xy \cdot 1$ são equivalentes. O **complemento** da função booleana F é a função $\overline{F}$, em que $\overline{F}(x_1, \ldots, x_n) = F(x_1, \ldots, x_n)$. Sejam F e G funções booleanas de n variáveis. A **soma booleana** $F + G$ e o **produto booleano** FG são definidos por

$$(F + G)(x_1, \ldots, x_n) = F(x_1, \ldots, x_n) + G(x_1, \ldots, x_n),$$
$$(FG)(x_1, \ldots, x_n) = F(x_1, \ldots, x_n)G(x_1, \ldots, x_n).$$

TABELA 2

x	y	z	xy	$\bar{z}$	$F(x, y, z) = xy + \bar{z}$
1	1	1	1	0	1
1	1	0	1	1	1
1	0	1	0	0	0
1	0	0	0	1	1
0	1	1	0	0	0
0	1	0	0	1	1
0	0	1	0	0	0
0	0	0	0	1	1

TABELA 3 As Funções Booleanas de Duas Variáveis.

x	y	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Uma função booleana de duas variáveis é uma função de um conjunto de quatro elementos, a saber, pares de elementos de $B = \{0, 1\}$, para B, um conjunto com dois elementos. Portanto, existem 16 funções booleanas diferentes de duas variáveis. Na Tabela 3 mostramos os valores das 16 funções booleanas diferentes de duas variáveis, chamadas de $F_1, F_2, \ldots, F_{16}$.

EXEMPLO 7 Quantas funções booleanas diferentes de n variáveis existem?

Solução: A partir da regra do produto para contagem, segue que existem 2^n n-uplas diferentes de 0s e 1s. Como uma função booleana é uma associação de 0 ou 1 a cada uma destas 2^n n-uplas diferentes, a regra do produto mostra que existem 2^{2^n} funções booleanas diferentes de n variáveis. ◀

A Tabela 4 mostra o número de funções booleanas diferentes de uma a seis variáveis. O número dessas funções cresce extremamente rápido.

Identidades da Álgebra Booleana

Existem muitas identidades na álgebra booleana. As mais importantes estão mostradas na Tabela 5. Essas identidades são particularmente úteis na simplificação de projetos de circuitos. Cada uma das identidades na Tabela 5 pode ser demonstrada usando uma tabela. Demonstraremos uma das propriedades distributivas dessa maneira no Exemplo 8. As demonstrações das propriedades restantes são deixadas como exercício para o leitor.

EXEMPLO 8 Mostre que a propriedade distributiva $x(y + z) = xy + xz$ é válida.

Solução: A verificação dessa identidade é mostrada na Tabela 6. A identidade é válida porque as duas últimas colunas da tabela coincidem. ◀

TABELA 4 O Número de Funções Booleanas de n Variáveis.

Número de variáveis	*Número de funções*
1	4
2	16
3	256
4	65.536
5	4.294.967.296
6	18.446.744.073.709.551.616

TABELA 5 Identidades Booleanas.	
Identidade	***Nome***
$\bar{\bar{x}} = x$	Propriedade do duplo complemento
$x + x = x$ $x \cdot x = x$	Propriedades idempotentes
$x + 0 = x$ $x \cdot 1 = x$	Propriedades dos elementos neutros
$x + 1 = 1$ $x \cdot 0 = 0$	Propriedades de dominação
$x + y = y + x$ $xy = yx$	Propriedades comutativas
$x + (y + z) = (x + y) + z$ $x(yz) = (xy)z$	Propriedades associativas
$x + yz = (x + y)(x + z)$ $x(y + z) = xy + xz$	Propriedades distributivas
$\overline{(xy)} = \bar{x} + \bar{y}$ $\overline{(x + y)} = \bar{x}\,\bar{y}$	Leis de De Morgan
$x + xy = x$ $x(x + y) = x$	Propriedades de absorção
$x + \bar{x} = 1$	Propriedade da unidade
$x\bar{x} = 0$	Propriedade do zero

O leitor deveria comparar as identidades booleanas da Tabela 5 com as equivalências lógicas da Tabela 6 da Seção 1.2 e com o conjunto de identidades da Tabela 1 da Seção 2.2. Todas são casos especiais do mesmo conjunto de identidades em uma estrutura abstrata mais geral. Cada coleção de identidades pode ser obtida fazendo as traduções adequadas. Por exemplo, podemos transformar cada uma das identidades da Tabela 5 em uma equivalência lógica transformando cada variável booleana em uma variável proposicional, cada 0 em um **F**, cada 1 em um **V**, cada soma booleana em uma disjunção, cada produto booleano em uma conjunção e cada complementação em uma negação, como ilustraremos no Exemplo 9.

TABELA 6 Verificando Uma das Propriedades Distributivas.							
x	y	z	$y + z$	xy	xz	$x(y + z)$	$xy + xz$
1	1	1	1	1	1	1	1
1	1	0	1	1	0	1	1
1	0	1	1	0	1	1	1
1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0

EXEMPLO 9 Traduza a propriedade distributiva $x + yz = (x + y)\,(x + z)$ da Tabela 5 para uma equivalência lógica.

Solução: Para traduzir uma identidade booleana para uma equivalência lógica, trocamos cada variável booleana por uma variável proposicional. Aqui, mudaremos as variáveis booleanas x, y e z para variáveis proposicionais p, q e r. A seguir, mudamos cada soma booleana para uma disjunção e cada produto booleano para uma conjunção. (Observe que 0 e 1 não aparecem nesta identidade e que a complementação também não aparece.) Isso transforma a identidade booleana na equivalência lógica

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r).$$

Essa equivalência lógica é uma das propriedades distributivas da lógica proposicional da Tabela 6 da Seção 1.2. ◀

As identidades na álgebra booleana podem ser usadas para demonstrar mais identidades. Ilustraremos isso no Exemplo 10.

EXEMPLO 10 Demonstre a **propriedade de absorção** $x\,(x + y) = x$ usando as outras identidades da álgebra booleana mostradas na Tabela 5. (Isso é chamado de propriedade de absorção, pois absorver $x + y$ em x deixa x invariante.)

Exemplos Extras

Solução: Os passos usados para deduzir essa identidade e a propriedade usada em cada passo são os seguintes:

$x(x + y) = (x + 0)(x + y)$	Propriedade do elemento neutro para a soma booleana
$= x + 0 \cdot y$	Propriedade distributiva da soma booleana em relação ao produto booleano
$= x + y \cdot 0$	Propriedade comutativa para o produto booleano
$= x + 0$	Propriedade de dominação para o produto booleano
$= x.$	Propriedade do elemento neutro para a soma booleana

◀

Dualidade

As identidades na Tabela 5 vêm aos pares (exceto pela propriedade do duplo complemento e pelas propriedades da unidade e do zero.) Para explicar a relação entre as duas identidades de cada par, usaremos o conceito de dual. O **dual** de uma expressão booleana é obtido trocando as somas booleanas e os produtos booleanos e trocando os 0s e os 1s.

EXEMPLO 11 Encontre os duais de $x(y + 0)$ e de $\overline{x} \cdot 1 + (\overline{y} + z)$.

Solução: A troca dos sinais de $\cdot$ e dos sinais de $+$ e a troca dos 0s e dos 1s nessas duas expressões fornece seus duais. Os duais são $x + (y \cdot 1)$ e $(\overline{x} + 0)(\overline{y}z)$, respectivamente. ◀

O dual de uma função booleana F representada por uma expressão booleana é a função representada pelo dual dessa expressão. Essa função dual, indicada por F^d, não depende da expressão booleana particular usada para representar F. Uma identidade entre funções representadas por expressões booleanas permanece válida quando são tomados os duais em ambos os lados da identidade. (Veja o Exercício 30 para a razão pela qual isso é verdade.) Esse resultado, chamado de **princípio da dualidade**, é usado para obter novas identidades.

EXEMPLO 12 Construa uma identidade a partir da propriedade de absorção $x(x + y) = x$ considerando os duais.

Solução: Considerar os duais de ambos os lados dessa identidade produz a identidade $x + xy = x$, que também é chamada de propriedade de absorção e está mostrada na Tabela 5. ◀

A Definição Abstrata de uma Álgebra Booleana

Nesta seção, nos concentramos em funções e expressões booleanas. Entretanto, os resultados que estabelecemos podem ser traduzidos em resultados sobre proposições ou resultados sobre conjuntos. Por causa disso, é útil definir álgebras booleanas abstratamente. Uma vez que tenha sido mostrado que uma estrutura particular é uma álgebra booleana, então todos os resultados estabelecidos sobre álgebras booleanas em geral se aplicam a esta estrutura particular.

As álgebras booleanas podem ser definidas de diversas maneiras. A maneira mais comum é especificar as propriedades que as operações devem satisfazer, como é feito na Definição 1.

DEFINIÇÃO 1 Uma *álgebra booleana* é um conjunto B com duas operações binárias $\vee$ e $\wedge$, elementos 0 e 1 e uma operação unária $^-$ tal que estas propriedades sejam válidas para todo x, y e z em B:

$$\left.\begin{aligned} x \vee 0 &= x \\ x \wedge 1 &= x \end{aligned}\right\} \text{Propriedades dos elementos neutros}$$

$$\left.\begin{aligned} x \vee \bar{x} &= 1 \\ x \wedge \bar{x} &= 0 \end{aligned}\right\} \text{Propriedades dos complementares}$$

$$\left.\begin{aligned} (x \vee y) \vee z &= x \vee (y \vee z) \\ (x \wedge y) \wedge z &= x \wedge (y \wedge z) \end{aligned}\right\} \text{Propriedades associativas}$$

$$\left.\begin{aligned} x \vee y &= y \vee x \\ x \wedge y &= y \wedge x \end{aligned}\right\} \text{Propriedades comutativas}$$

$$\left.\begin{aligned} x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \\ x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) \end{aligned}\right\} \text{Propriedades distributivas}$$

Usando as leis dadas na Definição 1, é possível demonstrar muitas outras propriedades que são válidas para toda álgebra booleana, tal como as propriedades idempotente e de dominação. (Veja os exercícios 35 a 42.)

De nossa discussão prévia, $B = \{0, 1\}$, com as operações *OU* e *E* e o operador de complemento, satisfaz todas essas propriedades. O conjunto de todas as proposições em n variáveis, com os operadores $\vee$ e $\wedge$, com **F** e **V** e com o operador de negação, também satisfaz todas as propriedades de uma álgebra booleana, como pode ser visto da Tabela 6 da Seção 1.2. Analogamente, o conjunto dos subconjuntos de um conjunto universo U com as operações de união e intersecção, com o conjunto vazio e o conjunto universo, e com o operador de complemento de conjunto, é uma álgebra booleana como pode ser visto consultando a Tabela 1 da Seção 2.2. Assim, para estabelecer resultados sobre cada um entre expressões booleanas, proposições e conjuntos, precisamos apenas demonstrar resultados sobre álgebras booleanas abstratas.

As álgebras booleanas também podem ser definidas usando a noção de reticulados, discutida no Capítulo 8. Lembre-se de que um reticulado L é um conjunto parcialmente ordenado no qual todo par de elementos x e y tem um menor limitante superior, indicado por $\sup(x, y)$ e um maior limitante inferior, indicado por $\inf(x, y)$. Dados dois elementos x e y de L, podemos definir duas operações $\vee$ e $\wedge$ nos pares de elementos de L por $x \vee y = \sup(x, y)$ e $x \wedge y = \inf(x, y)$.

Para um reticulado L ser uma álgebra booleana como especificado na Definição1, ele deve ter duas propriedades. Primeiro, ele deve ser **complementado**. Para um reticulado ser complementado, ele deve ter um menor elemento 0 e um maior elemento 1 e para todo elemento x da rede deve existir um elemento $\bar{x}$, tal que $x \vee \bar{x} = 1$ e $x \wedge \bar{x} = 0$. Segundo, ele deve ser **distributivo**. Isso significa que, para todo x, y e z em L, $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ e $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$. Mostrar que um reticulado complementado e distributivo é uma álgebra booleana foi deixado como Exercício 39, no final desta seção.

Exercícios

1. Encontre os valores destas expressões.

a) $1 \cdot \bar{0}$ **b)** $1 + \bar{1}$ **c)** $\bar{0} \cdot 0$ **d)** $\overline{(1+0)}$

2. Encontre os valores, caso existam, da variável booleana x que satisfazem estas equações.

a) $x \cdot 1 = 0$ **b)** $x + x = 0$

c) $x \cdot 1 = x$ **d)** $x \,.\, \bar{x} = 1$

3. a) Mostre que $(1 \cdot 1) + (\overline{0 \cdot 1} + 0) = 1$.

b) Traduza a equação da parte (a) em uma equivalência proposicional, trocando cada 0 por um **F**, cada 1 por um **V**, cada soma booleana por uma disjunção, cada produto booleano por uma conjunção, cada complementação por uma negação e os sinais de igual por um sinal de equivalência proposicional.

4. a) Mostre que $(\overline{1 \cdot 0}) + (1 \cdot \bar{0}) = 1$.

b) Traduza a equação da parte (a) em uma equivalência proposicional, trocando cada 0 por um **F**, cada 1 por um **V**, cada soma booleana por uma disjunção, cada produto booleano por uma conjunção, cada complementação por uma negação e os sinais de igual por um sinal de equivalência proposicional.

5. Use uma tabela para expressar os valores de cada uma destas funções booleanas.

a) $F(x, y, z) = \bar{x}y$

b) $F(x, y, z) = x + yz$

c) $F(x, y, z) = x\bar{y} + \overline{(xyz)}$

d) $F(x, y, z) = x(yz + \bar{y}\bar{z})$

6. Use uma tabela para expressar os valores de cada uma destas funções booleanas.

a) $F(x, y, z) = \bar{z}$

b) $F(x, y, z) = \bar{x}y + \bar{y}z$

c) $F(x, y, z) = x\bar{y}z + \overline{(xyz)}$

d) $F(x, y, z) = \bar{y}(xz + \bar{x}\bar{z})$

7. Use um 3-cubo Q_3 para representar cada uma das funções booleanas do Exercício 5, colocando um círculo preto em cada vértice que corresponde a um 3-upla na qual esta função tem o valor 1.

8. Use um 3-cubo Q_3 para representar cada uma das funções booleanas do Exercício 6, colocando um círculo preto em cada vértice que corresponde a um 3-upla na qual esta função tem o valor 1.

9. Quais valores das variáveis booleanas x e y satisfazem $xy = x + y$?

10. Quantas funções booleanas diferentes de 7 variáveis existem?

11. Demonstre a propriedade de absorção $x + xy = x$ usando as outras propriedades da Tabela 5.

☞**12.** Mostre que $F(x, y, z) = xy + xz + yz$ tem o valor 1 se e somente se pelo menos duas das variáveis x, y e z tiverem o valor 1.

13. Mostre que $x\bar{y} + y\bar{z} + \bar{x}z = \bar{x}y + \bar{y}z + x\bar{z}$.

Os exercícios 14 a 23 tratam da álgebra booleana $\{0, 1\}$ com adição, multiplicação e complemento definidos no início da seção. Em cada caso, use uma tabela, como no Exemplo 8.

14. Verifique a propriedade do duplo complemento.

15. Verifique as propriedades idempotentes.

16. Verifique as propriedades dos elementos neutros.

17. Verifique as propriedades de dominação.

18. Verifique as propriedades comutativas.

19. Verifique as propriedades associativas.

20. Verifique a primeira propriedade distributiva da Tabela 5.

21. Verifique as leis de De Morgan.

22. Verifique a propriedade da unidade.

23. Verifique a propriedade do zero.

O operador booleano $\oplus$, chamado de operador *XOU*, é definido por $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$ e $0 \oplus 0 = 0$.

24. Simplifique estas expressões.

a) $x \oplus 0$ **b)** $x \oplus 1$

c) $x \oplus x$ **d)** $x \oplus \bar{x}$

25. Mostre que estas identidades são válidas.

a) $x \oplus y = (x + y)\overline{(xy)}$

b) $x \oplus y = (x\bar{y}) + (\bar{x}y)$

26. Mostre que $x \oplus y = y \oplus x$.

27. Demonstre ou mostre que não são válidas estas igualdades.

a) $x \oplus (y \oplus z) = (x \oplus y) \oplus z$

b) $x + (y \oplus z) = (x + y) \oplus (x + z)$

c) $x \oplus (y + z) = (x \oplus y) + (x \oplus z)$

28. Encontre os duais destas expressões booleanas.

a) $x + y$ **b)** $\bar{x}\,\bar{y}$

c) $xyz + \bar{x}\,\bar{y}\,\bar{z}$ **d)** $x\bar{z} + x \cdot 0 + \bar{x} \cdot 1$

***29.** Suponha que F seja uma função booleana representada por uma expressão booleana nas variáveis $x_1, \ldots, x_n$. Mostre que $F^d(x_1, \ldots, x_n) = \overline{F(\bar{x}_1, \ldots, \bar{x}_n)}$.

***30.** Mostre que, se F e G forem funções booleanas representadas por expressões booleanas em n variáveis e $F = G$, então $F^d = G^d$, em que F^d e G^d são as funções booleanas representadas pelos duais das expressões booleanas que representam F e G, respectivamente. [*Dica:* Use o resultado do Exercício 29.]

***31.** Quantas funções booleanas diferentes $F(x, y, z)$ existem, tal que $F(\bar{x}, \bar{y}, \bar{z}) = F(x, y, z)$ para todos os valores das variáveis booleanas x, y e z?

***32.** Quantas funções booleanas diferentes $F(x, y, z)$ existem, tal que $F(\bar{x}, y, z) = F(x, \bar{y}, z) = F(x, y, \bar{z})$ para todos os valores das variáveis booleanas x, y e z?

33. Mostre que você obtém as leis de De Morgan para proposições (na Tabela 5 da Seção 1.2) quando você transforma as leis de De Morgan para álgebra booleana da Tabela 6 em equivalências lógicas.

34. Mostre que você obtém as propriedades de absorção para proposições (na Tabela 5 da Seção 1.2) quando transforma as propriedades de absorção para álgebra booleana da Tabela 5 em equivalências lógicas.

Nos exercícios 35 a 42, use as propriedades da Definição 1 para mostrar que as propriedades enunciadas são válidas para toda álgebra booleana.

35. Mostre que em uma álgebra booleana as **propriedades idempotentes** $x \vee x = x$ e $x \wedge x = x$ são válidas para todo elemento x.

36. Mostre que em uma álgebra booleana todo elemento x tem um único complemento $\bar{x}$ tal que $x \vee \bar{x} = 1$ e $x \wedge \bar{x} = 0$.

37. Mostre que em uma álgebra booleana o complemento do elemento 0 é o elemento 1 e vice-versa.

38. Demonstre que em uma álgebra booleana a **propriedade do duplo complemento** é válida; isto é, $\bar{\bar{x}} = x$ para todo elemento x.

39. Mostre que as **leis de De Morgan** são válidas em uma álgebra booleana. Isto é, mostre que para todo x e y, $\overline{(x \vee y)} = \bar{x} \wedge \bar{y}$ e $\overline{(x \wedge y)} = \bar{x} \vee \bar{y}$.

40. Mostre que em uma álgebra booleana, as **propriedades modulares** são válidas. Isto é, mostre que $x \wedge (y \vee (x \wedge z)) = (x \wedge y) \vee (x \wedge z)$ e $x \vee (y \wedge (x \vee z)) = (x \vee y) \wedge (x \vee z)$.

41. Mostre que em uma álgebra booleana, se $x \vee y = 0$, então $x = 0$ e $y = 0$ e que, se $x \wedge y = 1$, então $x = 1$ e $y = 1$.

42. Mostre que em uma álgebra booleana, o **dual** de uma identidade, obtido trocando os operadores $\vee$ e $\wedge$ e trocando os elementos 0 e 1, também é uma identidade.

43. Mostre que um reticulado distributivo complementado é uma álgebra booleana.

11.2 Representação de Funções Booleanas

Dois problemas importantes da álgebra booleana serão estudados nesta seção. O primeiro problema é: dados os valores de uma função booleana, como pode ser encontrada uma expressão booleana que representa esta função? Este problema será resolvido mostrando que qualquer função booleana pode ser representada por uma soma booleana de produtos booleanos de variáveis e seus complementos. A solução deste problema mostra que toda função booleana pode ser representada usando os três operadores booleanos $\cdot$, $+$ e $^-$. O segundo problema é: existe um conjunto menor de operadores que pode ser usado para representar todas as funções booleanas? Responderemos este problema mostrando que todas as funções booleanas podem ser representadas usando apenas um operador. Ambos os problemas têm importância prática no projeto de circuitos.

Expansões em Somas de Produtos

Usaremos exemplos para ilustrar uma maneira importante de encontrar uma expressão booleana que represente uma função booleana.

EXEMPLO 1 Encontre expressões booleanas que representem as funções $F(x, y, z)$ e $G(x, y, z)$, que são dadas na Tabela 1.

TABELA 1

x	y	z	F	G
1	1	1	0	0
1	1	0	0	1
1	0	1	1	0
1	0	0	0	0
0	1	1	0	0
0	1	0	0	1
0	0	1	0	0
0	0	0	0	0

Solução: Uma expressão que tenha o valor 1 quando $x = z = 1$ e $y = 0$ e o valor 0, caso contrário, é necessária para representar F. Essa expressão pode ser formada considerando o produto booleano de x, $\bar{y}$ e z. Este produto, $x\bar{y}z$, tem o valor 1 se e somente se $x = \bar{y} = z = 1$, que vale se e somente se $x = z = 1$ e $y = 0$.

Para representar G, precisamos de uma expressão que seja igual a 1 quando $x = y = 1$ e $z = 0$ ou quando $x = z = 0$ e $y = 1$. Podemos formar uma expressão com esses valores considerando a soma booleana de dois produtos booleanos diferentes. O produto booleano $xy\bar{z}$ tem o valor 1 se e somente se $x = y = 1$ e $z = 0$. Analogamente, o produto $\bar{x}y\bar{z}$ tem o valor 1 se e somente se $x = z = 0$ e $y = 1$. A soma booleana destes dois produtos, $xy\bar{z} + \bar{x}y\bar{z}$, representa G, pois tem o valor 1 se e somente se $x = y = 1$ e $z = 0$ ou $x = z = 0$ e $y = 1$. ◀

O Exemplo 1 ilustra um procedimento para construir uma expressão booleana que represente uma função com valores dados. Cada combinação de valores das variáveis para a qual a função tenha o valor 1 leva a um produto booleano das variáveis ou seus complementos.

DEFINIÇÃO 1 Um *literal* é uma variável booleana ou seu complemento. Um *mintermo* das variáveis booleanas $x_1, x_2, \ldots, x_n$ é um produto booleano $y_1y_2 \ldots y_n$, no qual $y_i = x_i$ ou $y_i = \bar{x}_i$. Portanto, um mintermo é um produto de n literais, com um literal para cada variável.

Um mintermo tem o valor 1 para uma e só uma combinação de valores de suas variáveis. Mais precisamente, o mintermo $y_1y_2 \ldots y_n$ é 1 se e somente se cada y_i é 1, e isto ocorre se e somente se $x_i = 1$ quando $y_i = x_i$ e $x_i = 0$ quando $y_i = \bar{x}_i$.

EXEMPLO 2 Encontre um mintermo que seja igual a 1 se $x_1 = x_3 = 0$ e $x_2 = x_4 = x_5 = 1$ e igual a 0, caso contrário.

Solução: O mintermo $\bar{x}_1x_2\bar{x}_3x_4x_5$ tem o conjunto correto de valores. ◀

Considerando somas booleanas de mintermos distintos, podemos construir expressões booleanas com um conjunto especificado de valores. Em particular, uma soma booleana de mintermos tem o valor 1 quando exatamente um dos mintermos na soma tem o valor 1. Ela tem o valor 0 para todas as outras combinações de valores das variáveis. Conseqüentemente, dada uma função booleana, pode ser formada uma soma booleana de mintermos que tem o valor 1 quando esta função booleana tiver o valor 1, e tem o valor 0 quando a função tiver o valor 0. Os mintermos nesta soma booleana correspondem àquelas combinações de valores para as quais a função tem o valor 1. A soma de mintermos que representa a função é chamada de **expansão em soma de produtos** ou **forma normal disjuntiva** da função booleana. (Veja o Exercício 42 na Seção 1.2 para o desenvolvimento da forma normal disjuntiva no cálculo proposicional.)

Links

EXEMPLO 3 Encontre a expansão em soma de produtos da função $F(x, y, z) = (x + y)\bar{z}$.

Exemplos Extras

Solução: Encontraremos a expansão em soma de produtos de $F(x, y, z)$ de duas maneiras. Primeiro, usaremos identidades booleanas para expandir o produto e simplificar. Encontramos que

$$\begin{aligned}
F(x, y, z) &= (x + y)\bar{z} \\
&= x\bar{z} + y\bar{z} && \text{Propriedade distributiva} \\
&= x1\bar{z} + 1y\bar{z} && \text{Propriedade do elemento neutro} \\
&= x(y + \bar{y})\bar{z} + (x + \bar{x})y\bar{z} && \text{Propriedade da unidade} \\
&= xy\bar{z} + x\bar{y}\bar{z} + xy\bar{z} + \bar{x}y\bar{z} && \text{Propriedade distributiva} \\
&= xy\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z}. && \text{Propriedade idempotente}
\end{aligned}$$

Segundo, podemos construir uma expansão em soma de produtos determinando os valores de F para todos os possíveis valores das variáveis x, y e z. Esses valores são encontrados na Tabela 2. A expansão em soma de produtos de F é a soma booleana de três mintermos que correspondem às três linhas desta tabela que dão o valor 1 para a função. Isto dá

$$F(x, y, z) = xy\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z}.$$ ◀

Também é possível encontrar uma expressão booleana que represente uma função booleana considerando o produto booleano de somas booleanas. A expansão resultante é chamada de **forma normal conjuntiva** ou **expansão em produto de somas** da função. Essas expansões podem ser encontradas a partir das expansões em somas de produtos considerando os duais. Como encontrar essas expansões diretamente está descrito no Exercício 10 no final desta seção.

TABELA 2					
x	y	z	$x+y$	$\bar{z}$	$(x+y)\bar{z}$
1	1	1	1	0	0
1	1	0	1	1	1
1	0	1	1	0	0
1	0	0	1	1	1
0	1	1	1	0	0
0	1	0	1	1	1
0	0	1	0	0	0
0	0	0	0	1	0

Completude Funcional

Toda função booleana pode ser expressa como uma soma booleana de mintermos. Cada mintermo é o produto booleano de variáveis booleanas ou seus complementos. Isso mostra que toda função booleana pode ser representada usando os operadores booleanos $\cdot$, $+$ e $^-$. Como todas as funções booleanas podem ser representadas usando esses operadores, dizemos que o conjunto $\{\cdot, +, ^-\}$ é **funcionalmente completo**. Podemos achar um conjunto menor de operadores funcionalmente completos? Podemos fazer isso se um dos três operadores deste conjunto puder ser expresso em termos dos outros dois. Isso pode ser feito usando uma das leis de De Morgan. Podemos eliminar todas as somas booleanas usando a identidade

$$x + y = \bar{\bar{x}}\,\bar{\bar{y}},$$

que é obtida considerando complementos em ambos os lados na segunda lei de De Morgan, dada na Tabela 5 da Seção 11.1, e então aplicando a propriedade do duplo complemento. Isso significa que o conjunto $\{\cdot, ^-\}$ é funcionalmente completo. Analogamente, poderíamos eliminar todos os produtos booleanos usando a identidade

$$xy = \overline{\bar{x} + \bar{y}},$$

que é obtida considerando os complementos de ambos os lados da primeira lei de De Morgan, dada na Tabela 5 da Seção 11.1, e então aplicando a propriedade do duplo complemento. Conseqüentemente, $\{+, ^-\}$ é funcionalmente completo. Observe que o conjunto $\{+, \cdot\}$ não é funcionalmente completo, pois é impossível expressar a função booleana $F(x) = \bar{x}$ usando esses operadores (veja o Exercício 19).

Encontramos conjuntos com dois operadores que são funcionalmente completos. Podemos encontrar um conjunto menor de operadores funcionalmente completos, ou seja, um conjunto com apenas um operador? Esses conjuntos existem. Defina dois operadores, o operador $|$ ou ***NE***, definido por $1 \mid 1 = 0$ e $1 \mid 0 = 0 \mid 1 = 0 \mid 0 = 1$; e o operador $\downarrow$ ou ***NOU***, definido por $1 \downarrow 1 = 1 \downarrow 0 = 0 \downarrow 1 = 0$ e $0 \downarrow 0 = 1$. Ambos os conjuntos $\{\mid\}$ e $\{\downarrow\}$ são funcionalmente completos. Para ver que $\{\mid\}$ é funcionalmente completo, como $\{\cdot, ^-\}$ é funcionalmente completo, tudo o que temos a fazer é mostrar que ambos os operadores $\cdot$ e $^-$ podem ser expressos usando apenas o operador $|$. Isso é feito da seguinte maneira

$$\bar{x} = x \mid x,$$
$$xy = (x \mid y) \mid (x \mid y).$$

O leitor deve verificar estas identidades (veja o Exercício 14). Deixamos a demonstração de que $\{\downarrow\}$ é funcionalmente completo para o leitor (veja os exercícios 15 e 16).

Exercícios

1. Encontre um produto booleano das variáveis booleanas x, y e z, ou seus complementos, que tenha o valor 1 se e somente se

a) $x = y = 0, z = 1$. **b)** $x = 0, y = 1, z = 0$.
c) $x = 0, y = z = 1$. **d)** $x = y = z = 0$.

2. Encontre as expansões em somas de produtos destas funções booleanas.

a) $F(x, y) = \bar{x} + y$ **b)** $F(x, y) = x\bar{y}$
c) $F(x, y) = 1$ **d)** $F(x, y) = \bar{y}$

3. Encontre as expansões em somas de produtos destas funções booleanas.

a) $F(x, y, z) = x + y + z$
b) $F(x, y, z) = (x + z)y$
c) $F(x, y, z) = x$
d) $F(x, y, z) = x\bar{y}$

4. Encontre as expansões em somas de produtos da função booleana $F(x, y, z)$ que é igual a 1 se e somente se

a) $x = 0$. **b)** $xy = 0$.
c) $x + y = 0$. **c)** $xyz = 0$.

5. Encontre as expansões em somas de produtos da função booleana $F(w, x, y, z)$ que tem o valor 1 se e somente se um número ímpar entre w, x, y e z tem o valor 1.

6. Encontre as expansões em somas de produtos da função booleana $F(x_1, x_2, x_3, x_4, x_5)$ que tem o valor 1 se e somente se três ou mais das variáveis x_1, x_2, x_3, x_4 e x_5 têm o valor 1.

Uma outra maneira de encontrar uma expressão booleana que representa uma função booleana é formar um produto booleano de somas de literais. Os exercícios 7 a 11 se referem a representações desse tipo.

7. Encontre uma soma booleana que contenha x ou $\bar{x}$, y ou $\bar{y}$ e z ou $\bar{z}$ e que tenha o valor 0 se e somente se

a) $x = y = 1, z = 0$. **b)** $x = y = z = 0$.
c) $x = z = 0, y = 1$.

8. Encontre um produto booleano de somas booleanas de literais que tenha o valor 0 se e somente se ou $x = y = 1$ e $z = 0$, ou $x = z = 0$ e $y = 1$, ou $x = y = z = 0$. [*Dica*: Considere o produto booleano das somas booleanas encontradas nas partes (a), (b) e (c) do Exercício 7.]

9. Mostre que a soma booleana $y_1 + y_2 + \cdots + y_n$, em que $y_i = x_i$ ou $y_i = \bar{x}_i$, tem o valor 0 para exatamente uma combinação de valores das variáveis, a saber, quando $x_i = 0$ se $y_i = x_i$ e $x_i = 1$ se $y_i = \bar{x}_i$. Esta soma booleana é chamada de **maxtermo**.

10. Mostre que uma função booleana pode ser representada como um produto booleano de maxtermos. Esta representação é chamada de **expansão em produto de somas ou forma normal conjuntiva** da função. [*Dica:* Inclua um maxtermo neste produto para cada combinação das variáveis para as quais a função tem o valor 0.]

11. Encontre a expansão em produto de somas de cada uma das funções booleanas do Exercício 3.

12. Expresse cada uma destas funções booleanas usando os operadores $\cdot$ e $^-$.

a) $x + y + z$ **b)** $x + \bar{y}(\bar{x} + z)$
c) $\overline{(x + \bar{y})}$ **d)** $\bar{x}(x + \bar{y} + \bar{z})$

13. Expresse cada uma das funções booleanas do Exercício 12 usando os operadores $+$ e $^-$.

14. Mostre que

a) $\bar{x} = x \mid x$. **b)** $xy = (x \mid y) \mid (x \mid y)$.
c) $x + y = (x \mid x) \mid (y \mid y)$.

15. Mostre que

a) $\bar{x} = x \downarrow x$.
b) $xy = (x \downarrow x) \downarrow (y \downarrow y)$.
c) $x + y = (x \downarrow y) \downarrow (x \downarrow y)$.

16. Mostre que $\{\downarrow\}$ é funcionalmente completo usando o Exercício 15.

17. Expresse cada uma das funções booleanas do Exercício 3 usando o operador $\mid$.

18. Expresse cada uma das funções booleanas do Exercício 3 usando o operador $\downarrow$.

19. Mostre que o conjunto de operadores $\{+, \cdot\}$ não é funcionalmente completo.

20. Estes conjuntos de operadores são funcionalmente completos?

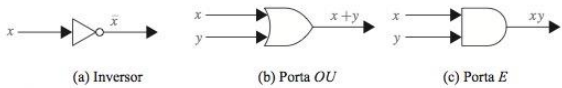
a) $\{+, \oplus\}$ **b)** $\{^-, \oplus\}$ c) $\{\cdot, \oplus\}$

11.3 Portas Lógicas

Introdução

Links

Álgebra booleana é usada para modelar o esquema de circuito de aparelhos eletrônicos. Cada entrada e cada saída desse aparelho pode ser considerada como um membro do conjunto $\{0, 1\}$. Um computador, ou outros aparelhos eletrônicos, é feito de muitos circuitos. Cada circuito pode ser projetado usando as regras da álgebra booleana que foram estudadas nas Seções 11.1 e 11.2. Os elementos básicos dos circuitos são chamados de **portas**. Cada tipo de porta implementa uma operação booleana. Nesta seção, definimos diversos tipos de portas. Usando essas portas, aplicaremos as regras da álgebra booleana para projetar circuitos que façam diversas tarefas. Os circuitos que estudaremos neste capítulo dão saídas que dependem apenas da entrada e não de seu



(a) Inversor (b) Porta *OU* (c) Porta *E*

FIGURA 1 Tipos Básicos de Portas.

FIGURA 2 Portas com *n* Entradas.

estado atual. Em outras palavras, esses circuitos não têm capacidade de memória. Eles são chamados de **circuitos combinatórios** ou **redes de portas**.

Vamos construir circuitos combinatórios usando três tipos de elementos. O primeiro é um **inversor**, que aceita o valor de uma variável booleana como entrada e produz o complemento desse valor como sua saída. O símbolo usado para um inversor é mostrado na Figura 1(a). A entrada do inversor está mostrada no lado esquerdo, entrando no elemento, e a saída está mostrada no lado direito, saindo do elemento.

O próximo tipo de elemento que usaremos é a **porta *OU***. As entradas dessa porta são os valores de duas ou mais variáveis booleanas. A saída é a soma booleana de seus valores. O símbolo usado para uma porta *OU* está mostrado na Figura 1(b). As entradas da porta *OU* estão mostradas no lado esquerdo, entrando no elemento, e a saída está mostrada no lado direito, saindo do elemento.

O terceiro tipo de elemento que usaremos é a **porta *E***. As entradas dessa porta são os valores de duas ou mais variáveis booleanas. A saída é o produto booleano de seus valores. O símbolo usado para uma porta *E* está mostrado na Figura 1(c). As entradas da porta *E* estão mostradas no lado esquerdo, entrando no elemento, e a saída está mostrada no lado direito, saindo do elemento.

Permitiremos entradas múltiplas para as portas *E* e *OU*. As entradas para cada uma dessas portas estão mostradas no lado esquerdo, entrando no elemento, e a saída está mostrada no lado direito. Exemplos de portas *E* e *OU* com *n* entradas estão mostrados na Figura 2.

Combinações de Portas

Circuitos combinatórios podem ser construídos usando uma combinação de inversores, portas *OU* e portas *E*. Quando são formadas combinações de circuitos, algumas portas podem compartilhar entradas. Isso é mostrado em uma de duas maneiras na descrição de circuitos. Um método é usar ramificações que indicam todas as portas que usam uma dada entrada. O outro método é indicar essa entrada separadamente para cada porta. A Figura 3 ilustra as duas maneiras de mostrar portas com os mesmos valores de entrada. Observe também que a saída de uma porta pode ser usada como entrada por um outro elemento ou mais, como mostrado na Figura 3. Ambos os desenhos na Figura 3 descrevem o circuito que produz a saída $xy + \bar{x}y$.

EXEMPLO 1 Construa circuitos que produzam as seguintes saídas: (a) $(x + y)\bar{x}$, (b) $\bar{x}\,\overline{(y + \bar{z})}$ e (c) $(x + y + z)(\bar{x}\,\bar{y}\,\bar{z})$.

Solução: Circuitos que produzem essas saídas são mostrados na Figura 4. ◀

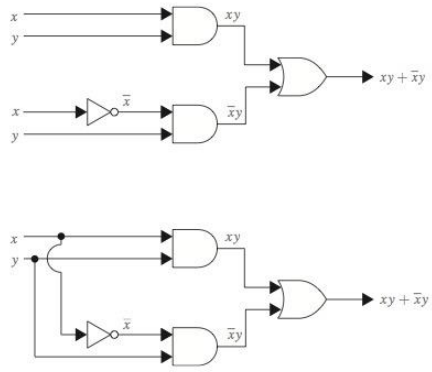


FIGURA 3 **Duas Maneiras de Desenhar o Mesmo Circuito.**

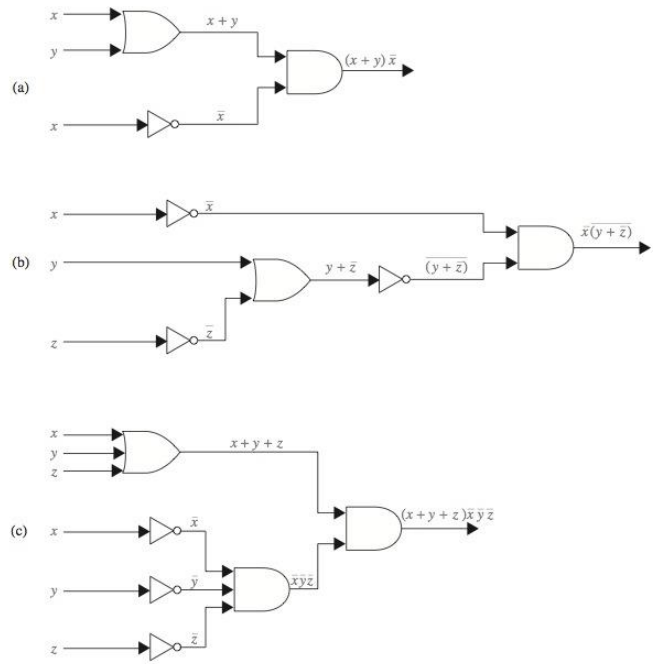


FIGURA 4 **Circuitos que Produzem as Saídas Especificadas no Exemplo 1.**

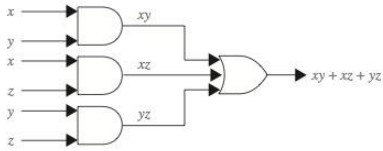


FIGURA 5 **Circuito para Votação Majoritária.**

Exemplos de Circuitos

Daremos alguns exemplos de circuitos que executam algumas funções úteis.

EXEMPLO 2 Um comitê de três indivíduos decide questões para uma organização. Cada indivíduo vota sim ou não para cada proposta que aparece. Uma proposta é aprovada se ela recebe pelo menos dois votos sim. Projete um circuito que determina se a proposta foi aprovada.

Exemplos Extras

Solução: Seja $x = 1$ se o primeiro indivíduo votar sim e $x = 0$ se este indivíduo votar não; seja $y = 1$ se o segundo indivíduo votar sim e $y = 0$ se este indivíduo votar não; seja $z = 1$ se o terceiro indivíduo votar sim e $z = 0$ se este indivíduo votar não. Então, deve ser projetado um circuito que produza a saída 1 a partir das entradas x, y e z quando dois ou mais entre x, y e z forem 1. Uma representação da função booleana que tem esses valores de saída é $xy + xz + yz$ (veja o Exercício 12 na Seção 11.1). O circuito que implementa esta função está mostrado na Figura 5. ◀

EXEMPLO 3 Algumas instalações luminosas são controladas por mais de um interruptor. É necessário projetar circuitos de modo que, movendo qualquer um dos interruptores da instalação, a luz seja acesa quando estiver apagada e vice-versa. Projete circuitos que façam isso quando existirem dois interruptores e quando existirem três interruptores.

Solução: Começaremos projetando o circuito que controla a instalação luminosa quando são usados dois interruptores diferentes. Seja $x = 1$ quando o primeiro interruptor estiver fechado e $x = 0$ quando estiver aberto e seja $y = 1$ quando o segundo interruptor estiver fechado e $y = 0$ quando estiver aberto. Seja $F(x, y) = 1$ quando a luz estiver acesa e $F(x, y) = 0$ quando estiver apagada. Podemos decidir arbitrariamente que a luz estará acesa quando ambos os interruptores estiverem fechados, de modo que $F(1, 1) = 1$. Isso determina todos os outros valores de F. Quando um dos dois interruptores é aberto, a luz apaga, de modo que $F(1, 0) = F(0, 1) = 0$. Quando o outro interruptor também for aberto, a luz acende, de modo que $F(0, 0) = 1$. A Tabela 1 mostra esses valores. Vemos que $F(x, y) = xy + \bar{x}\,\bar{y}$. Essa função é implementada pelo circuito mostrado na Figura 6.

TABELA 1

x	y	$F(x, y)$
1	1	1
1	0	0
0	1	0
0	0	1

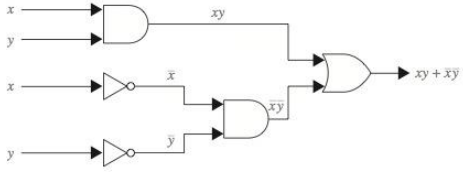


FIGURA 6 **Circuito para uma Lâmpada Controlada por Dois Interruptores.**

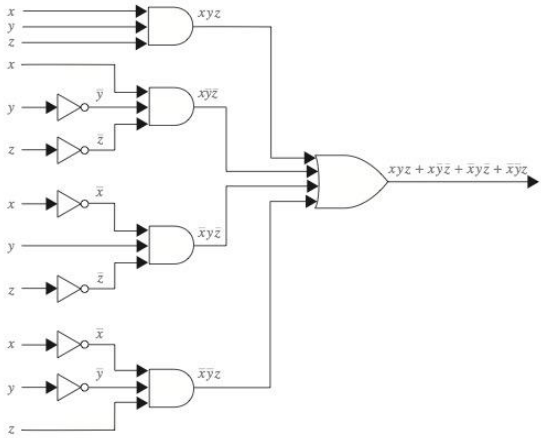


FIGURA 7 **Circuito para uma Instalação Controlada por Três Interruptores.**

TABELA 2

x	*y*	*z*	*F(x, y, z)*
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

Projetaremos agora um circuito para três interruptores. Sejam x, y e z as variáveis booleanas que indicam se cada um dos três interruptores está fechado. Consideramos $x = 1$ quando o primeiro interruptor estiver fechado e $x = 0$ quando ele abrir; $y = 1$ quando o segundo interruptor estiver fechado e $y = 0$ quando ele abrir; e $z = 1$ quando o terceiro interruptor estiver fechado e $z = 0$ quando ele abrir. Seja $F(x, y, z) = 1$ quando a luz estiver acesa e $F(x, y, z) = 0$ quando a luz estiver apagada. Podemos especificar arbitrariamente que a luz esteja acesa quando todos os três interruptores estiverem fechados, de modo que $F(1, 1, 1) = 1$. Isso determina todos os outros valores de F. Quando um interruptor é aberto, a luz apaga, de modo que $F(1, 1, 0) = F(1, 0, 1) = F(0, 1, 1) = 0$. Quando um segundo interruptor é aberto, a luz acende, de modo que $F(1, 0, 0) = F(0, 1, 0) = F(0, 0, 1) = 1$. Finalmente, quando o terceiro interruptor é aberto, a luz apaga novamente, de modo que $F(0, 0, 0) = 0$. A Tabela 2 mostra os valores desta função.

A função F pode ser representada pela sua expansão em soma de produtos como $F(x, y, z) = xyz + x\bar{y}\bar{z} + \bar{x}y\bar{z} + \bar{x}\bar{y}z$. O circuito mostrado na Figura 7 implementa esta função. ◀

Links

Somadores

TABELA 3
Entrada e Saída para o Meio Somador.

Entrada		*Saída*	
x	*y*	*s*	*c*
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

Ilustraremos como os circuitos lógicos podem ser usados para executar a adição de dois inteiros positivos a partir de suas expansões binárias. Construiremos um circuito para fazer esta adição a partir de alguns circuitos componentes. Primeiro, construiremos um circuito que pode ser usado para encontrar $x + y$, em que x e y são dois bits. A entrada de nosso circuito será x e y, pois cada um deles tem o valor 0 ou o valor 1. A saída consistirá em dois bits, a saber, s e c, em que s é o bit da soma e c é o bit do "vai um". Esse circuito é chamado de **circuito de saída múltipla**, pois tem mais de uma saída. O circuito que estamos projetando é chamado de **meio somador**, pois ele soma dois bits, sem considerar o "vai um" de uma adição anterior. Mostramos a entrada e a saída do meio somador na Tabela 3, na qual vemos que $c = xy$ e que $s = x\bar{y} + \bar{x}y = (x + y)\overline{(xy)}$. Portanto, o circuito mostrado na Figura 8 calcula o bit soma s e o bit "vai um" c para os bits x e y.

Usamos o **somador completo** para calcular o bit soma e o bit "vai um" quando são somados dois bits e um "vai um". As entradas do somador completo são os bits x e y e o "vai um" c_i. As saídas são o bit soma s e o novo "vai um" c_{i+1}. As entradas e as saídas de um somador completo são exibidas na Tabela 4.

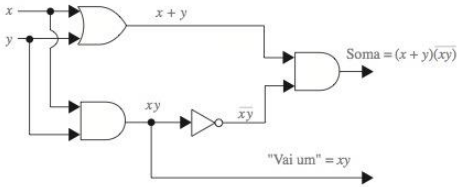


FIGURA 8 O Meio Somador.

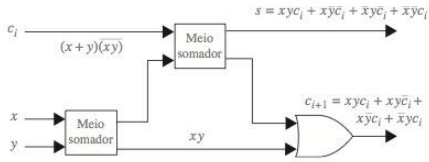


FIGURA 9 Somador Completo.

TABELA 4
Entrada e Saída para o Somador Completo.

Entrada			*Saída*	
x	y	c_i	s	$c_i + 1$
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

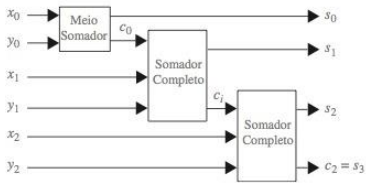


FIGURA 10 Somando Dois Inteiros de Três Bits com Meio Somador e Somadores Completos.

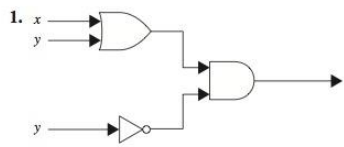
As duas saídas do somador completo, o bit soma s e o bit "vai um" c_{i+1}, são dados pelas expansões em soma de produtos $xyc_i + x\bar{y}\bar{c}_i + \bar{x}y\bar{c}_i + \bar{x}\bar{y}\,c_i$ e $xyc_i + xy\bar{c}_i + x\bar{y}c_i + \bar{x}yc_i$, respectivamente. Entretanto, em vez de projetar o somador completo a partir do zero, usaremos meios somadores para produzir a saída desejada. Um circuito para o somador completo usando meios somadores é mostrado na Figura 9.

Finalmente, na Figura 10 mostramos como meios somadores e somadores completos podem ser usados para somar dois inteiros de três bits $(x_2x_1x_0)_2$ e $(y_2y_1y_0)_2$ para produzir a soma $(s_3s_2s_1s_0)_2$. Observe que s_3, o bit de ordem mais alta na soma, é dado pelo "vai um" c_2.

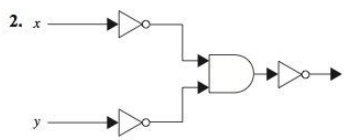
Exercícios

Nos exercícios 1 a 5, encontre a saída dos circuitos dados.

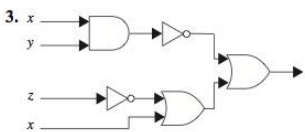
1.



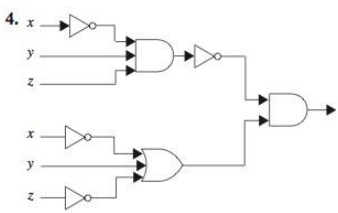
2.



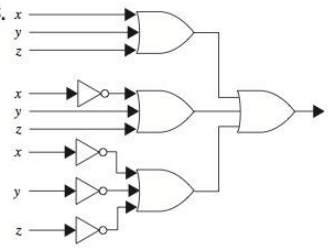
3.



4.



5.



6. Construa, a partir de inversores, portas E e portas OU, circuitos que produzam estas saídas.
a) $\bar{x} + y$ **b)** $(x + y)\,x$
c) $xyz + \bar{x}\,\bar{y}\,\bar{z}$ **d)** $(\bar{x} + z)(y + \bar{z})$

7. Projete um circuito que implemente a votação majoritária para cinco indivíduos.

8. Projete um circuito para uma instalação luminosa controlada por quatro interruptores, no qual mover um dos interruptores acende a luz quando ela estiver apagada e a apaga quando estiver acesa.

9. Mostre como a soma de dois inteiros de cinco bits pode ser encontrada usando meio somadores e somadores completos.

10. Construa um circuito para um meio subtrator usando portas E, portas OU e inversores. Um **meio subtrator** tem dois bits como entrada e produz, como saída, um bit de diferença e um "empresta um".

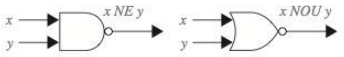
11. Construa um circuito para um subtrator completo usando portas E, portas OU e inversores. Um **subtrator completo** tem dois bits e um bit "empresta um" como entrada e produz, como saída, um bit de diferença e um "empresta um".

12. Use os circuitos dos exercícios 10 e 11 para encontrar a diferença entre dois inteiros de quatro bits, em que o primeiro inteiro é maior do que o segundo inteiro.

***13.** Construa um circuito que compare os inteiros de dois bits $(x_1x_0)_2$ e $(y_1y_0)_2$, retornando uma saída 1 quando o primeiro desses números for maior e uma saída 0 caso contrário.

***14.** Construa um circuito que calcule o produto dos inteiros de dois bits $(x_1x_0)_2$ e $(y_1y_0)_2$. O circuito deve ter quatro saídas para os bits no produto.

Duas portas que são usadas com freqüência em circuitos são as portas NE e NOU. Quando as portas NE ou NOU são usadas para representar circuitos, nenhum outro tipo de porta é necessário. A notação para essas portas são as seguintes:



***15.** Use portas NE para construir circuitos com estas saídas.
a) $\bar{x}$ **b)** $x + y$ **c)** xy **d)** $x \oplus y$

***16.** Use portas NOU para construir circuitos para as saídas dadas no Exercício 15.

***17.** Construa um meio somador usando portas NE.

***18.** Construa um meio somador usando portas NOU.

Um **multiplexer** é um circuito de chaves que produz como saída um entre um conjunto de bits de entrada com base no valor de bits de controle.

19. Construa um multiplexer usando portas E, portas OU e inversores que têm como entrada os quatro bits x_0, x_1, x_2 e x_3 e os dois bits de controle c_0 e c_1. Monte o circuito de modo que x_i seja a saída, em que i é o valor do inteiro de dois bits $(c_1c_0)_2$.

A **profundidade** de um circuito combinatório pode ser definida especificando que a profundidade da entrada inicial é 0 e se uma porta tem n entradas em profundidades $d_1, d_2, \ldots, d_n$, respectivamente, então suas saídas têm profundidade igual a $\max(d_1, d_2, \ldots, d_n) + 1$; este valor também é definido como a profundidade da porta. A profundidade de um circuito combinatório é a profundidade máxima das portas no circuito.

20. Encontre a profundidade
a) do circuito construído no Exemplo 2 para a votação majoritária entre três pessoas.
b) o circuito construído no Exemplo 3 para a lâmpada controlada por dois interruptores.
c) o meio somador mostrado na Figura 8.
d) o somador completo mostrado na Figura 9.

11.4 Minimização de Circuitos

Introdução

A eficiência de um circuito combinatório depende do número e da disposição de suas portas. O processo de projetar um circuito combinatório começa com a tabela que especifica a saída para cada combinação de valores de entrada. Podemos sempre usar a expansão em soma de produtos de um circuito para encontrar um conjunto de portas lógicas que implementarão este circuito. Entretanto, a expansão em soma de produtos pode conter muito mais termos do que o necessário. Os termos em uma expansão em soma de produtos que diferem em apenas uma variável, de modo que em um termo esta variável ocorre e no outro termo ocorre o complemento desta variável, podem ser combinados. Por exemplo, considere o circuito que tem a saída 1 se e somente se $x = y = z = 1$ ou $x = z = 1$ e $y = 0$. A expansão em soma de produtos desse circuito é $xyz + x\bar{y}z$.

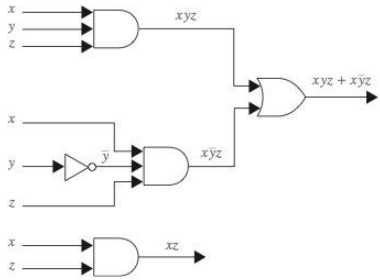


FIGURA 1 Dois Circuitos com a Mesma Saída.

Os dois produtos nesta expansão diferem em exatamente uma variável, a saber, y. Eles podem ser combinados como

$$\begin{aligned} xyz + x\bar{y}z &= (y + \bar{y})\,(xz) \\ &= 1 \cdot (xz) \\ &= xz. \end{aligned}$$

Logo, xz é uma expressão booleana com menos operadores que representa o circuito. Mostramos duas implementações diferentes desse circuito na Figura 1. O segundo circuito usa apenas uma porta, enquanto o primeiro circuito usa três portas e um inversor.

Esse exemplo mostra que combinar termos na expansão em soma de produtos de um circuito leva a uma expressão mais simples para o circuito. Descreveremos dois procedimentos que simplificam essas expansões.

O objetivo de ambos os procedimentos é produzir somas booleanas de produtos booleanos que representem uma função com o mínimo de produtos de literais, de modo que esses produtos contenham o menor número possível de literais entre todas as somas de produtos que representam a função booleana. Encontrar essa soma de produtos é chamado de **minimização de uma função booleana**, que torna possível construir um circuito para essa função que usa o menor número de portas e o menor número de entradas para as portas *E* e para as portas *OU* no circuito, entre todos os circuitos para a expressão booleana que está sendo minimizada.

Até o início de década de 1960, portas lógicas eram componentes individuais. Para reduzir custos, era importante usar o menor número possível de portas para produzir o resultado desejado. Entretanto, em meados da década de 1960, foi desenvolvida a tecnologia de circuitos integrados que tornou possível combinar portas em um único chip. Mesmo que atualmente seja possível construir circuitos integrados cada vez mais complexos em chips de baixo custo, a minimização de funções booleanas continua importante.

Reduzir o número de portas em um chip pode levar a circuitos mais confiáveis e reduzir o custo de produção do chip. Além disso, a minimização torna possível colocar mais circuitos no mesmo chip. Ela também reduz o número de entradas das portas de um circuito, que reduz o tempo usado por um circuito para computar sua saída. Além do mais, o número de entradas de uma porta pode ser limitado por causa da tecnologia particular usada para construir portas lógicas.

O primeiro procedimento que introduziremos, conhecido como mapas de Karnaugh (ou mapas K), foi desenvolvido na década de 1950 para ajudar a minimizar circuitos manualmente. Os mapas K são úteis na minimização de circuitos com até seis variáveis, embora eles se tornem bem complexos mesmo para cinco ou seis variáveis. O segundo procedimento que descreveremos, o método de Quine–McCluskey, foi inventado na década de 1960. Ele automatiza o processo de minimizar circuitos combinatórios e pode ser implementado como um programa de computador.

Infelizmente, minimizar funções booleanas com muitas variáveis é um problema computacionalmente intensivo. Foi mostrado que esse é um problema NP-completo (veja a Seção 3.3 e [Ka93]), de modo que a existência de algoritmos polinomiais no tempo para minimizar circuitos booleanos é pouco provável. O método de Quine–McCluskey tem complexidade exponencial. Na prática, ele pode ser usado apenas quando o número de literais não exceder 10. Desde a década de 1970, foram desenvolvidos diversos novos algoritmos para minimizar circuitos combinatórios (veja [Ha93] e [Ka93]). No entanto, com os melhores algoritmos já desenvolvidos, apenas circuitos com não mais de 25 variáveis podem ser minimizados. Além disso, métodos heurísticos (ou práticos) podem ser usados para simplificar substancialmente, mas não necessariamente minimizar, expressões booleanas com um número maior de literais.

Mapas de Karnaugh

Links

Para reduzir o número de termos em uma expressão booleana que representa um circuito, é necessário encontrar termos para combinar. Existe um método gráfico, chamado de **mapa de Karnaugh** ou **mapa K**, para encontrar termos para combinar nas funções booleanas que envolvem um número relativamente pequeno de variáveis. O método que descreveremos foi introduzido por Maurice Karnaugh, em 1953. Seu método baseia-se em trabalhos anteriores de E. W. Veitch. (Este método é, em geral, aplicado apenas quando a função envolve seis variáveis ou menos.) Os mapas K nos dão um método visual para simplificar expansões em soma de produtos; eles não são adequados para mecanizar este processo. Ilustraremos primeiro como os mapas K podem ser usados para simplificar expansões de funções booleanas em duas variáveis. Continuaremos mostrando como eles podem ser usados para minimizar funções booleanas em três variáveis e, então, em quatro variáveis. A seguir, descreveremos os conceitos que podem ser usados para estendê-los para minimizar funções booleanas em mais de quatro variáveis.

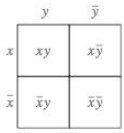


FIGURA 2
Mapas K em Duas Variáveis.

Existem quatro mintermos possíveis em expansões em soma de produtos de uma função booleana em duas variáveis x e y. Um mapa K para uma função booleana nessas duas variáveis consiste em quatro células, nas quais é colocado um 1 na célula que representa um mintermo se este mintermo está presente na expansão. As células são denominadas **adjacentes** se os mintermos que elas representam diferem em exatamente um literal. Por exemplo, a célula que representa $\bar{x}y$ é adjacente às células que representam xy e $\bar{x}\bar{y}$. As quatro células e os termos que elas representam estão mostrados na Figura 2.

EXEMPLO 1 Encontre mapas K para (a) $x\,y + \bar{x}\,y$, (b) $x\,\bar{y} + \bar{x}\,y$ e (c) $x\,\bar{y} + \bar{x}\,y + \bar{x}\,\bar{y}$.

Solução: Incluímos um 1 na célula quando o mintermo representado por ela está presente na expansão em soma de produtos. Os três mapas K estão mostrados na Figura 3. ◀

Podemos identificar mintermos que podem ser combinados a partir do mapa K. Sempre que existirem 1s em duas células adjacentes no mapa K, os mintermos representados por essas células podem ser combinados em um produto que envolve apenas uma das variáveis. Por exemplo, $x\,\bar{y}$ e $\bar{x}\,\bar{y}$ são representados por células adjacentes e podem ser combinados em $\bar{y}$, pois $x\,\bar{y} + \bar{x}\,\bar{y} = (x + \bar{x})\bar{y} = \bar{y}$. Além disso, se existirem 1s nas quatro células, os quatro mintermos podem ser combinados em um termo, a saber, a expressão booleana 1, que não envolve nenhuma das variáveis. Circulamos blocos de células no mapa K que representam mintermos que podem ser combinados e então encontramos a soma de produtos correspondente. O objetivo é identificar os

MAURICE KARNAUGH (NASCIDO EM 1924) Maurice Karnaugh, nascido na cidade de Nova York, fez sua graduação na City College of New York e seu mestrado e doutorado na Universidade de Yale. Ele foi um membro da equipe tecnológica no Bell Laboratories, de 1952 até 1966, e Gerente de Pesquisa e Desenvolvimento na Federal Systems Division da AT&T, de 1966 a 1970. Em 1970, ingressou na IBM como membro da equipe de pesquisa. Karnaugh fez contribuições fundamentais às aplicações das técnicas digitais tanto na computação quanto nas telecomunicações. Seus interesses atuais incluem sistemas que se baseiam em conhecimento em computação e métodos de busca heurísticos.

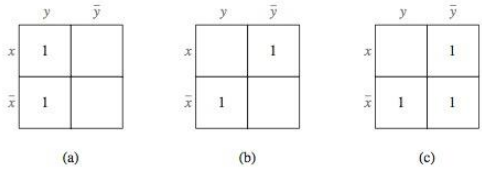


FIGURA 3 **Mapas K para as Expansões em Soma de Produtos no Exemplo 1.**

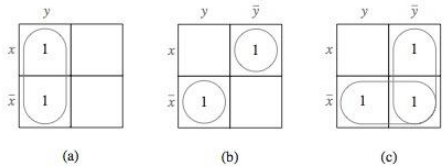


FIGURA 4 **Simplificação das Expansões em Soma de Produtos do Exemplo 2.**

os maiores blocos possíveis e cobrir todos os 1s com o menor número de blocos usando os blocos maiores primeiro e sempre usando os maiores blocos possíveis.

EXEMPLO 2 Simplifique as expansões em soma de produtos dadas no Exemplo1.

Solução: O agrupamento de mintermos é mostrado na Figura 4, usando os mapas K para essas expansões. As expansões mínimas para essas somas de produtos são (a) y, (b) $x\,\bar{y} + \bar{x}\,y$ e (c) $\bar{x} + \bar{y}$. ◀

Um mapa K em três variáveis é um retângulo dividido em oito células. Essas células representam os oito mintermos possíveis em três variáveis. Duas células são denominadas adjacentes se os mintermos que elas representam diferem em exatamente um literal. Uma das maneiras de formar um mapa K em três variáveis é mostrada na Figura 5(a). Este mapa K pode ser considerado como estando em um cilindro, conforme mostrado na Figura 5(b). No cilindro, duas células têm uma fronteira em comum se e somente se elas forem adjacentes.

Para simplificar uma expansão em soma de produtos em três variáveis, usamos o mapa K para identificar blocos de mintermos que podem ser combinados. Blocos de duas células adjacentes representam pares de mintermos que podem ser combinados em um produto de duas literais; blocos 2×2 e 4×1 de células representam mintermos que podem ser combinados em um único literal; e o bloco de todas as oito células representa um produto sem nenhum literal, ou

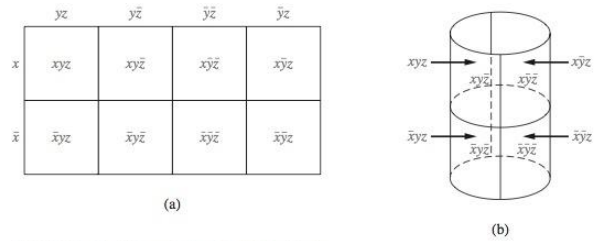


FIGURA 5 **Mapas K em Três Variáveis.**

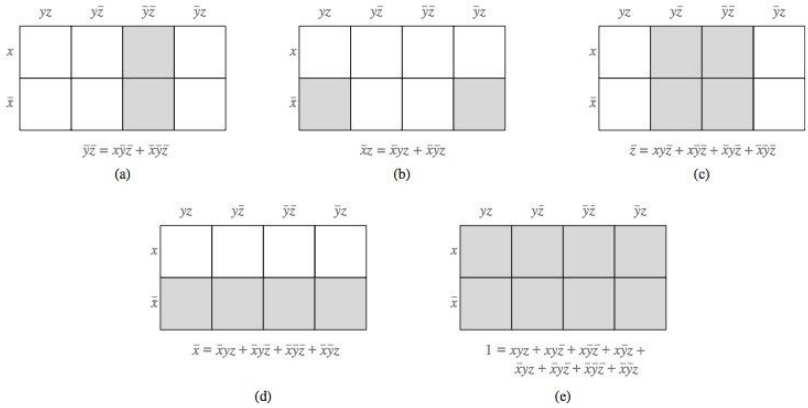


FIGURA 6 Blocos em Mapas K em Três Variáveis.

seja, a função 1. Na Figura 6, são mostrados os blocos 1×2, 2×1, 2×2, 4×1 e 4×2 e os produtos que eles representam.

O produto de literais que correspondem a um bloco só de 1s é chamado de **implicante** da função que está sendo minimizada. Ele é chamado de **implicante primo** se esse bloco de 1s não estiver contido em um bloco maior de 1s que representa o produto de menos literais que neste produto.

O objetivo é identificar os maiores blocos possíveis no mapa e cobrir todos os 1s no mapa com o menor número de blocos, usando os blocos maiores primeiro. Os maiores blocos possíveis são sempre escolhidos, mas podemos sempre escolher um bloco se ele for o único de 1s que cobre um 1 no mapa K. Esse bloco representa um **implicante primo essencial**. Cobrindo todos os 1s no mapa com blocos que correspondem a implicantes primos, podemos expressar a soma de produtos como uma soma de implicantes primos. Observe que pode existir mais de uma maneira de cobrir todos os 1s usando o menor número possível de blocos.

O Exemplo 3 ilustra como os mapas K em três variáveis são usados.

EXEMPLO 3 Use mapas K para minimizar estas expansões em soma de produtos.

(a) $xy\bar{z} + x\bar{y}\,\bar{z} + \bar{x}\,y\,z + \bar{x}\,\bar{y}\,\bar{z}$

(b) $x\bar{y}z + x\bar{y}\,\bar{z} + \bar{x}\,y\,z + \bar{x}\,\bar{y}\,z + \bar{x}\,\bar{y}\,\bar{z}$

(c) $xyz + x\,y\,\bar{z} + x\,\bar{y}\,z + x\,\bar{y}\,\bar{z} + \bar{x}\,y\,z + \bar{x}\,\bar{y}\,z + \bar{x}\,\bar{y}\,\bar{z}$

(d) $x\,y\,\bar{z} + x\,\bar{y}\,\bar{z} + \bar{x}\,\bar{y}\,z + \bar{x}\,\bar{y}\,\bar{z}$

Solução: Os mapas K para essas expansões em soma de produtos são mostrados na Figura 7. O agrupamento de blocos mostra que expansões mínimas em somas booleanas de produtos boolea-nos são (a) $x\bar{z} + \bar{y}\,\bar{z} + \bar{x}yz$, (b) $\bar{y} + \bar{x}z$, (c) $x + \bar{y} + z$ e (d) $x\,\bar{z} + \bar{x}\,\bar{y}$. Na parte (d), observe que os implicantes primos $x\bar{z}$ e $\bar{x}\,\bar{y}$ são implicantes primos essenciais, mas o implicante primo $\bar{y}\,\bar{z}$ é um implicante primo que não é essencial, pois as células que ele cobre são cobertas por outros dois implicantes primos. ◀

Um mapa K em quatro variáveis é um quadrado que é dividido em 16 células. As células representam os 16 mintermos possíveis em 4 variáveis. Uma das maneiras de formar um mapa K em quatro variáveis é mostrada na Figura 8.

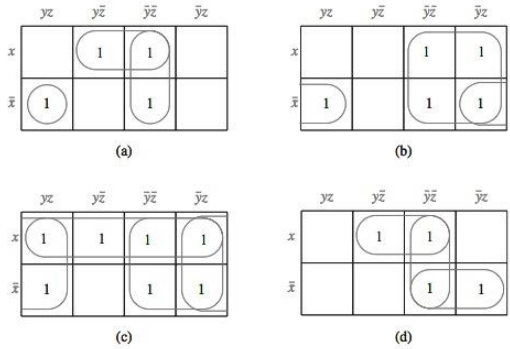


FIGURA 7 O Uso de Mapas K em Três Variáveis.

Duas células são adjacentes se e somente se os mintermos que elas representam diferem apenas em um literal. Conseqüentemente, cada célula é adjacente a quatro outras células. O mapa K de uma expansão em soma de produtos em quatro variáveis pode ser considerado como estando em um toro, de modo que células adjacentes tenham uma fronteira comum (veja o Exercício 28). A simplificação de uma expansão em soma de produtos em quatro variáveis é feita identificando aqueles blocos de 2, 4, 8 ou 16 células que representam mintermos que podem ser combinados. Cada célula que representa um mintermo deve ou ser utilizada para formar um produto usando menos literais ou ser incluída na expansão. Na Figura 9, são mostrados alguns exemplos de blocos que representam produtos de três literais, produtos de dois literais e um único literal.

Como no caso dos mapas K em duas e três variáveis, o objetivo é identificar os maiores blocos de 1s no mapa, que correspondem aos primos implicantes, e cobrir todos os 1s com o mínimo necessário de blocos, usando sempre os maiores blocos possíveis primeiro. O Exemplo 4 ilustra como os mapas K em quatro variáveis são usados.

EXEMPLO 4 Use mapas K para simplificar estas expansões em soma de produtos.

(a) $wxyz + wxy\bar{z} + wx\bar{y}\bar{z} + w\bar{x}yz + w\bar{x}\bar{y}z + w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}z + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z}$

(b) $wx\bar{y}\bar{z} + w\bar{x}yz + w\bar{x}y\bar{z} + w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z}$

(c) $wxy\bar{z} + wx\bar{y}\bar{z} + w\bar{x}yz + w\bar{x}y\bar{z} + w\bar{x}\bar{y}\bar{z} + \bar{w}xyz + \bar{w}xy\bar{z} + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z}$

	yz	$yz̄$	$ȳz̄$	$ȳz$
wx	$wxyz$	$wxyz̄$	$wxȳz̄$	$wxȳz$
$wx̄$	$wx̄yz$	$wx̄yz̄$	$wx̄ȳz̄$	$wx̄ȳz$
$w̄x̄$	$w̄x̄yz$	$w̄x̄yz̄$	$w̄x̄ȳz̄$	$w̄x̄ȳz$
$w̄x$	$w̄xyz$	$w̄xyz̄$	$w̄xȳz̄$	$w̄xȳz$

FIGURA 8 Mapas K em Quatro Variáveis.

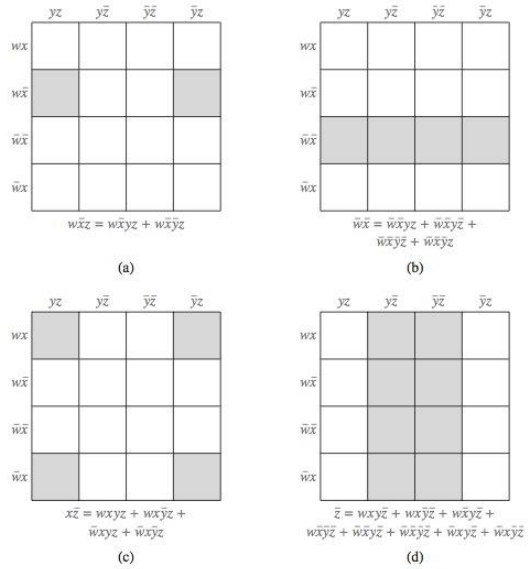


FIGURA 9 Blocos em Mapas K em Quatro Variáveis.

Solução: Os mapas K para essas expansões são exibidos na Figura 10. Usar os blocos mostrados leva à soma de produtos (a) $wyz + wx\bar{z} + w\bar{x}\bar{y} + \bar{w}\bar{x}y + \bar{w}x\bar{y}z$, (b) $\bar{y}\bar{z} + w\bar{x}y + \bar{x}\bar{z}$ e (c) $\bar{z} + \bar{w}x + w\bar{x}y$. O leitor deve determinar se existem outras escolhas de blocos em cada parte, que levem a somas de produtos diferentes que representem essas funções booleanas. ◀

Os mapas K podem ser usados realisticamente para minimizar funções booleanas com cinco ou seis variáveis, mas, além disso, eles raramente são usados, pois se tornam extremamente complicados. Entretanto, os conceitos usados nos mapas K desempenham papel importante em algoritmos mais novos. Além disso, dominar esses conceitos o ajudará a entender esses algoritmos mais novos e programas para projetos, com o auxílio do computador (CAD), que os implementam. À

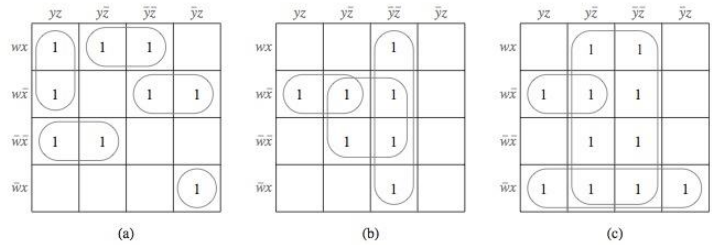


FIGURA 10 O Uso de Mapas K em Quatro Variáveis.

medida que desenvolvermos esses conceitos, seremos capazes de ilustrá-los referindo-nos novamente a nossa discussão de minimização de funções booleanas em três e quatro variáveis.

Os mapas K que usamos para minimizar as funções booleanas em duas, três e quatro variáveis são construídos com retângulos 2×2, 2×4 e 4×4, respectivamente. Além disso, células correspondentes na linha de cima e na linha de baixo e na coluna mais à esquerda e na coluna mais à direita em cada um desses casos são consideradas adjacentes, pois representam mintermos que diferem em apenas um literal. Podemos construir mapas K para minimizar funções booleanas em mais de quatro variáveis de modo parecido. Usamos um retângulo com $2^{\lfloor n/2 \rfloor}$ linhas e $2^{\lceil n/2 \rceil}$ colunas. (Esses mapas K contêm 2^n células, porque $\lceil n/2 \rceil + \lfloor n/2 \rfloor = n$.) As linhas e as colunas precisam ser posicionadas de modo que as células que representam mintermos que difiram em apenas um literal sejam adjacentes ou sejam consideradas adjacentes pela especificação de adjacências adicionais para linhas e colunas. Para ajudar (mas não completamente) a alcançar isso, as linhas e as colunas de um mapa K são dispostas usando códigos de Gray (veja a Seção 9.5), em que associamos seqüências de bits e produtos, especificando que um 1 corresponde ao aparecimento de uma variável e um 0, ao aparecimento de seu complemento. Por exemplo, em um mapa K de dimensão 10, o código de Gray 01110 usado para rotular uma linha corresponde ao produto $\bar{x}_1 x_2 x_3 x_4 \bar{x}_5$.

EXEMPLO 5 Os mapas K que usamos para minimizar funções booleanas com quatro variáveis têm duas linhas e duas colunas. Ambas as linhas e as colunas são dispostas usando o código de Gray 11,10,00,01. As linhas representam produtos $w\,x$, $w\,\bar{x}$, $\bar{w}\,\bar{x}$ e $\bar{w}\,x$, respectivamente, e as colunas correspondem aos produtos yz, $y\bar{z}$, $\bar{y}\bar{z}$ e $\bar{y}z$, respectivamente. Usando códigos de Gray e as células adjacentes na primeira e na última linhas e na primeira e na última colunas, garantimos que os mintermos que diferem em apenas uma variável sejam sempre adjacentes. ◀

EXEMPLO 6 Para minimizar funções booleanas em cinco variáveis, usamos mapas K com $2^3 = 8$ colunas e $2^2 = 4$ linhas. Rotulamos as quatro linhas usando o código de Gray 11,10,00,01, que corresponde a x_1x_2, $x_1\bar{x}_2$, $\bar{x}_1\bar{x}_2$ e $\bar{x}_1x_2$, respectivamente. Rotulamos as oito colunas usando o código de Gray 111,110,100,101,001,000,010,011, que corresponde aos termos $x_3x_4x_5$, $x_3x_4\bar{x}_5$, $x_3\bar{x}_4\bar{x}_5$, $x_3\bar{x}_4x_5$, $\bar{x}_3\bar{x}_4x_5$, $\bar{x}_3\bar{x}_4\bar{x}_5$, $\bar{x}_3x_4\bar{x}_5$ e $\bar{x}_3x_4x_5$, respectivamente. Usar códigos de Gray para rotular colunas e linhas garante que os mintermos representados por células adjacentes difiram apenas em uma variável. Entretanto, para garantir que todas as células que representam produtos que diferem em apenas uma variável sejam consideradas adjacentes, consideramos células nas linhas de cima e de baixo como sendo adjacentes, bem como as células na primeira e na oitava colunas, na primeira e na quarta colunas, na segunda e na sétima colunas, na terceira e na sexta colunas e na quinta e na oitava colunas (como o leitor deve verificar). ◀

Para usar um mapa K para minimizar uma função booleana em n variáveis, primeiro desenhamos um mapa do tamanho apropriado. Colocamos 1s em todas as células que correspondam aos mintermos na expansão em soma de produtos dessa função. A seguir, identificamos todos os implicantes primos da função booleana. Para fazer isso, procuramos blocos que consistam em 2^k células agrupadas, todas com um 1, em que $1 \leq k \leq n$. Esses blocos correspondem ao produto de $n - k$ literais. (O Exercício 33 pede que o leitor verifique isso.) Além disso, um bloco de 2^k células cada uma com um 1 que não esteja contido em um bloco de 2^{k+1} células, cada uma das quais com um 1, representa um implicante primo. A razão de esse implicante ser um implicante primo é que nenhum produto obtido removendo um literal também é representado por um bloco de células, todas com 1s.

EXEMPLO 7 Um bloco de oito células que representa o produto de dois literais em um mapa K para minimizar funções booleanas em cinco variáveis, todas com 1s, é um implicante primo se ele não estiver contido em um bloco maior de 16 células, todas com 1s que representam um único literal. ◀

Uma vez que todos os implicantes primos tenham sido identificados, o objetivo é encontrar o menor subconjunto possível desses implicantes primos com a propriedade de que as células que os representam cobrem todas as células que contêm um 1 no mapa K. Começamos escolhendo

implicantes primos essenciais, pois cada um destes é representado por um bloco que cobre uma célula que contém um 1 e que não está coberta por nenhum outro implicante primo. Adicionamos implicantes primos para garantir que todos os 1s no mapa K sejam cobertos. Quando o número de variáveis é grande, este último passo pode se tornar extremamente complicado.

Condições Indiferentes (Don't care)

Em determinados circuitos, nos importamos apenas com a saída para algumas combinações de valores de entrada, pois as outras combinações de valores de entrada não são possíveis ou nunca ocorrem. Isso nos dá liberdade na produção de um circuito simples com a saída desejada, pois os valores de saída para todas as combinações que nunca ocorrem podem ser escolhidos arbitrariamente. Os valores da função para estas combinações são chamados de **condições indiferentes** (ou ***don't care***). Um *d* é usado em um mapa K para marcar aquelas combinações de valores das variáveis para as quais a função pode ser arbitrariamente definida. No processo de minimização, podemos associar 1s como valores àquelas combinações de valores de entrada que levem a blocos maiores no mapa K. Isso é ilustrado no Exemplo 8.

EXEMPLO 8 Uma maneira de codificar expansões decimais usando bits é usar os quatro bits da expansão binária de cada algarismo na expansão decimal. Por exemplo, 873 é codificado como 100001110011. Essa codificação de uma expansão decimal é chamada de **expansão decimal codificada binária**. Como existem 16 blocos de quatro bits e apenas 10 algarismos, existem seis combinações de quatro bits que não são usadas para codificar algarismos. Suponha que seja construído um circuito que produza uma saída de 1 se o algarismo decimal for maior que ou igual a 5 e uma saída 0 se o algarismo decimal for menor que 5. Como esse circuito pode ser construído de modo simples usando portas *OU*, portas *E* e inversores?

Solução: Indique por $F(w, x, y, z)$ a saída do circuito, em que *wxyz* é uma expansão binária de um algarismo decimal. Os valores de F estão mostrados na Tabela 1. O mapa K para F, com *d*s nas posições *don't care*, é exibido na Figura 11(a). Podemos ou incluir ou excluir quadrados com *d*s dos blocos. Isso nos dá muitas escolhas possíveis para os blocos. Por exemplo, excluir todos os quadrados com *d*s e formar blocos, como mostrado na Figura 11(b), produz a expressão $w\bar{x}\bar{y} + \bar{w}xy + \bar{w}xz$. Incluir alguns dos *d*s, excluir outros e formar blocos, como mostrado na Figura 11(c), produz a expressão $w\bar{x} + \bar{w}xy + x\bar{y}z$. Finalmente, incluir todos os *d*s e usar os blocos mostrados na Figura 11(d) produz a expansão em soma de produtos mais simples possível, a saber, $F(x, y, z) = w + xy + xz$. ◀

TABELA 1

Algarismo	*w*	*x*	*y*	*z*	*F*
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1

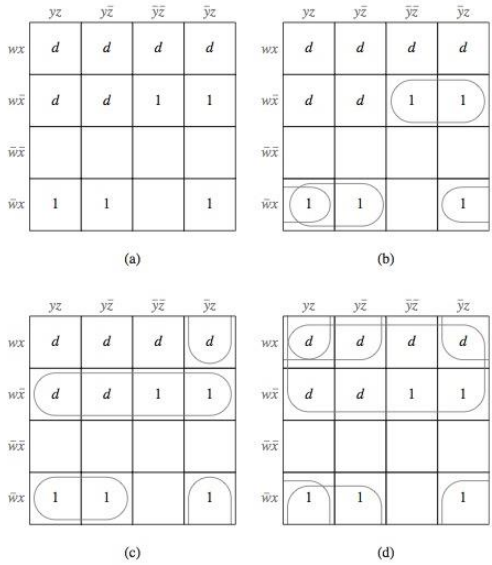


FIGURA 11 O Mapa K para *F* Mostrando Suas Posições *Don't Care*.

O Método de Quine–McCluskey

Links

Vimos que os mapas K podem ser usados para produzir expansões mínimas de funções booleanas, como somas booleanas de produtos booleanos. Entretanto, os mapas K são difíceis de usar quando existem mais de quatro variáveis. Além disso, o uso desses mapas baseia-se na inspeção visual para identificar os termos a serem agrupados. Por essas razões, existe uma necessidade de um procedimento para simplificar expansões em soma de produtos que possa ser mecanizado. O método de Quine–McCluskey é um desses procedimentos e pode ser usado para funções booleanas em qualquer número de variáveis. Ele foi desenvolvido na década de 1950, por W. V. Quine e E. J. McCluskey, Jr. Basicamente, o método Quine–McCluskey consiste em duas partes. A primeira parte encontra aqueles termos que são candidatos à inclusão em uma expansão mínima, como uma soma booleana de produtos booleanos. A segunda determina quais desses termos realmente usar. Usaremos o Exemplo 9 para ilustrar como, por combinações sucessivas de implicantes em implicantes com um literal a menos, esse procedimento funciona.

Links

EDWARD J. McCLUSKEY (NASCIDO EM 1929) Edward McCluskey freqüentou o Bowdoin College e o M.I.T., onde concluiu seu doutorado em engenharia elétrica, em 1956. Ele ingressou no Bell Telephone Laboratories em 1955, permanecendo lá até 1959. McCluskey foi professor de engenharia elétrica da Universidade de Princeton, de 1959 a 1966, tendo sido também diretor do Centro de Computação de Princeton, de 1961 a 1966. Em 1967, ele aceitou um cargo de professor de ciência da computação e engenharia elétrica na Universidade de Stanford, onde também exerceu o cargo de diretor do Laboratório de Sistemas Digitais, de 1969 a 1978. McCluskey trabalhou em diversas áreas da ciência da computação, incluindo computação com tolerância a falhas, arquitetura de computadores, verificação e projetos lógicos. Atualmente é diretor do Center for Reliable Computing na Universidade de Stanford. McCluskey também é membro da ACM.

TABELA 2		
Mintermo	*Seqüência de Bits*	*Número de 1s*
xyz	111	3
$x\bar{y}z$	101	2
$\bar{x}yz$	011	2
$\bar{x}\bar{y}z$	001	1
$\bar{x}\bar{y}\bar{z}$	000	0

EXEMPLO 9 Mostraremos como o método de Quine–McCluskey pode ser usado para encontrar uma expansão mínima equivalente a

$$xyz + x\bar{y}z + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}.$$

Representaremos os mintermos nesta expansão por seqüências de bits. O primeiro bit será 1, se x ocorrer, e 0, se $\bar{x}$ ocorrer. O segundo bit será 1, se y ocorrer, e 0, se $\bar{y}$ ocorrer. O terceiro bit será 1, se z ocorrer, e 0, se $\bar{z}$ ocorrer. Então, agruparemos esses termos de acordo com o número de 1s nas seqüências de bits correspondentes. Essa informação é exibida na Tabela 2.

Os mintermos que podem ser combinados são aqueles que diferem em exatamente um literal. Portanto, dois termos que podem ser combinados diferem por exatamente um no número de 1s nas seqüências de bits que os representam. Quando dois mintermos são combinados em um produto, esse produto contém dois literais. Um produto de dois literais é representado com um traço para indicar a variável que não ocorre. Por exemplo, os mintermos $x\bar{y}z$ e $\bar{x}\bar{y}z$, representados pelas seqüências de bits 101 e 001, podem ser combinados em $\bar{y}z$, representado pela seqüência -01. Todos os pares de mintermos que podem ser combinados e os produtos formados por essas combinações são exibidos na Tabela 3.

A seguir, todos os pares de produtos de dois literais que podem ser combinados são combinados em um literal. Dois desses produtos podem ser combinados se eles contiverem literais para as mesmas duas variáveis e os literais de apenas uma das variáveis diferirem. Em termos das seqüências que representam os produtos, estas devem ter um traço na mesma posição e devem diferir em exatamente uma das outras duas posições. Podemos combinar os produtos yz e $\bar{y}z$, representados pelas seqüências -11 e -01, em z, representado pela seqüência $--1$. Mostramos todas as combinações de termos que podem ser formadas dessa maneira na Tabela 3.

Links

WILLARD VAN ORMAN QUINE (1908–2000) Willard Quine, nascido em Akron, Ohio, freqüentou o Oberlin College e, mais tarde, a Universidade de Harvard, onde concluiu seu doutorado em filosofia, em 1932. Tornou-se um membro da sociedade Junior em Harvard em 1933, sendo nomeado para um cargo no corpo docente em 1936. Ele permaneceu em Harvard por toda sua vida profissional, exceto na Segunda Guerra Mundial, quando trabalhou para a marinha americana decifrando mensagens de submarinos alemães. Quine sempre se interessou por algoritmos, mas não por hardware. Ele chegou à descoberta do que é agora conhecido como método de Quine–McCluskey como um estratagema para ensinar lógica matemática, em vez de um método para simplificar circuitos de chaves. Quine foi um dos filósofos mais famosos do século XX. Ele fez contribuições fundamentais à teoria do conhecimento, lógica matemática e teoria dos conjuntos e à filosofia da lógica e da linguagem. Seus livros, incluindo *New Foundations of Mathematical Logic*, publicado em 1937, e *Word and Object*, publicado em 1960, tiveram um impacto profundo. Quine aposentou-se de Harvard em 1978, mas continuou a viajar entre sua casa em Beacon Hill e seu escritório, lá localizado. Ele usou a máquina de escrever Remington 1927, com a qual datilografou sua tese de doutorado, por toda sua vida. Ele chegou mesmo a realizar uma operação nessa máquina para adicionar uns poucos símbolos especiais, removendo o segundo ponto final, a segunda vírgula e o ponto de interrogação. Quando lhe perguntaram se ele sentia falta do ponto de interrogação, ele replicou, "Bem, você sabe, eu lido com certezas". Existe até mesmo a palavra *quine*, definida no *New Hacker's Dictionary* como um programa que gera uma cópia de seu próprio código-fonte como sua saída completa. Produzir o quine mais curto possível em uma linguagem de programação é um problema popular entre os hackers.

TABELA 3								
			Passo 1			*Passo 2*		
	Termo	*Seqüência de Bit*		*Termo*	*Seqüência*		*Termo*	*Seqüência*
1	xyz	111	(1,2)	xz	1−1	(1,2,3,4)	z	−−1
2	$x\bar{y}z$	101	(1,3)	yz	−11			
3	$\bar{x}yz$	011	(2,4)	$\bar{y}z$	−01			
4	$\bar{x}\bar{y}z$	001	(3,4)	$\bar{x}z$	0−1			
5	$\bar{x}\bar{y}\bar{z}$	000	(4,5)	$\bar{x}\bar{y}$	00−			

Na Tabela 3, indicamos também quais termos foram usados para formar produtos com menos literais; esses termos não serão necessários em uma expansão mínima. O próximo passo é identificar um conjunto mínimo de produtos necessários para representar a função booleana. Começamos com todos aqueles produtos que não foram usados para construir produtos com menos literais. A seguir, formamos a Tabela 4, que tem uma linha para cada produto candidato formado pela combinação dos termos originais e uma coluna para cada termo original; e colocamos um X na posição, se o termo original na expansão em soma de produtos foi usado para formar esse produto candidato. Nesse caso, dizemos que o produto candidato **cobre** o mintermo original. Precisamos incluir pelo menos um produto que cubra cada um dos mintermos originais. Conseqüentemente, sempre que existir apenas um X em uma coluna na tabela, deve ser usado o produto que corresponde à linha na qual este X está. A partir da Tabela 4, vemos que ambos z e $\bar{x}\,\bar{y}$ são necessários. Portanto, a resposta final é $z + \bar{x}\,\bar{y}$. ◀

Como foi ilustrado no Exemplo 9, o método Quine–McCluskey usa esta seqüência de passos para simplificar uma expressão em soma de produtos.

1. Expresse cada mintermo em n variáveis por uma seqüência de bits de comprimento n com um 1 na i-ésima posição, se x_i ocorrer e um 0 nesta posição, se $\bar{x}_i$ ocorrer.
2. Agrupe as seqüências de bits de acordo com o número de 1s presentes nelas.
3. Determine todos os produtos de $n - 1$ variáveis que podem ser formados considerando a soma booleana de mintermos na expansão. Mintermos que podem ser combinados são representados por seqüências de bits que diferem em exatamente uma posição. Represente esses produtos de $n - 1$ variáveis com seqüências que têm um 1 na i-ésima posição, se x_i ocorre, um 0 nesta posição, se $\bar{x}_i$ ocorrer e um traço nesta posição, se não existir nenhum literal que envolva x_i no produto.
4. Determine todos os produtos em $n - 2$ variáveis que podem ser formados considerando a soma booleana dos produtos de $n - 1$ variáveis encontrados no passo anterior. Os produtos em $n - 1$ variáveis que puderem ser combinados são representados por seqüências de bits que têm um traço na mesma posição e diferem em exatamente uma posição.
5. Continue combinando produtos booleanos em produtos de menos variáveis enquanto for possível.
6. Encontre todos os produtos booleanos que aparecem e que não foram usados para formar um produto booleano em um literal a menos.

TABELA 4					
	xyz	$x\bar{y}z$	$\bar{x}yz$	$\bar{x}\bar{y}z$	$\bar{x}\bar{y}\bar{z}$
z	X	X	X	X	
$\bar{x}\,\bar{y}$				X	X

TABELA 5		
Termo	*Seqüência de Bit*	*Número de 1s*
$wxy\bar{z}$	1110	3
$w\bar{x}yz$	1011	3
$\bar{w}xyz$	0111	3
$w\bar{x}y\bar{z}$	1010	2
$\bar{w}x\bar{y}z$	0101	2
$\bar{w}\bar{x}xyz$	0011	2
$\bar{w}\bar{x}\bar{y}z$	0001	1

7. Encontre o menor conjunto desses produtos booleanos, tal que a sua soma represente a função booleana. Isso é feito formando uma tabela que mostra quais mintermos são cobertos por quais produtos. Todo mintermo deve ser coberto por, pelo menos, um produto. O primeiro passo no uso dessa tabela é encontrar todos os implicantes primos essenciais. Cada implicante primo essencial deve ser incluído, pois ele é o único implicante primo que cobre um dos mintermos. Uma vez que você tenha encontrado os implicantes primos essenciais, podemos simplificar a tabela eliminando as colunas dos mintermos cobertos por esse implicante primo. Além disso, podemos eliminar qualquer implicante primo que cobre um subconjunto de mintermos cobertos por um outro implicante primo (como o leitor deve verificar). Além do mais, podemos eliminar da tabela a coluna para um mintermo, se existir outro mintermo que seja coberto por um subconjunto dos implicantes primos que cobrem este mintermo. Este processo de identificar implicantes primos essenciais que precisam ser incluídos, seguido pela eliminação dos implicantes primos redundantes e pela identificação dos mintermos que podem ser ignorados, é iterado até que a tabela não varie. Nesse ponto, usamos um procedimento de *backtracking* para encontrar a solução ótima, no qual adicionamos implicantes primos à cobertura para encontrar possíveis soluções, as quais comparamos com a melhor solução encontrada, até o momento, em cada passo.

Um último exemplo ilustrará como esse procedimento é usado para simplificar uma expansão em soma de produtos em quatro variáveis.

EXEMPLO 10 Use o método de Quine–McCluskey para simplificar a expansão em soma de produtos $wxy\bar{z} + w\bar{x}yz + w\bar{x}y\bar{z} + \bar{w}xyz + \bar{w}x\bar{y}z + \bar{w}\bar{x}xyz + \bar{w}\bar{x}\bar{y}z$.

TABELA 6								
			Passo 1			*Passo 2*		
	Termo	*Seqüência de Bits*		*Termo*	*Seqüência*		*Termo*	*Seqüência*
1	$wxy\bar{z}$	1110	(1,4)	$wy\bar{z}$	1–10	(3,5,6,7)	$\bar{w}z$	0––1
2	$w\bar{x}yz$	1011	(2,4)	$w\bar{x}y$	101–			
3	$\bar{w}xyz$	0111	(2,6)	$\bar{x}yz$	–011			
4	$w\bar{x}y\bar{z}$	1010	(3,5)	$\bar{w}xz$	01–1			
5	$\bar{w}x\bar{y}z$	0101	(3,6)	$\bar{w}yz$	0–11			
6	$\bar{w}\bar{x}xyz$	0011	(5,7)	$\bar{w}\bar{y}z$	0–01			
7	$\bar{w}\bar{x}\bar{y}z$	0001	(6,7)	$\bar{w}\bar{x}z$	00–1			

TABELA 7							
	$wxy\bar{z}$	$wx\bar{y}yz$	$\bar{w}xyz$	$wx\bar{y}\bar{z}$	$\bar{w}x\bar{y}z$	$\bar{w}\bar{x}yz$	$\bar{w}\bar{x}\bar{y}z$
$\bar{w}z$			X		X	X	X
$wy\bar{z}$	X			X			
$w\bar{x}y$		X		X			
$\bar{x}yz$		X				X	

Solução: Primeiro, representamos os mintermos por seqüências de bits e, então, agrupamos esses termos de acordo com o número de 1s nas seqüências de bits. Isso é mostrado na Tabela 5. Todos os produtos booleanos que podem ser formados considerando somas booleanas desses produtos são mostrados na Tabela 6.

Os únicos produtos que não foram usados para formar produtos em menos variáveis são $\bar{w}z$, $wy\bar{z}$, $w\bar{x}y$ e $\bar{x}yz$. Na Tabela 7, mostramos os mintermos cobertos por esses produtos. Para cobrir esses mintermos, devemos incluir $\bar{w}z$ e $wy\bar{z}$, pois esses produtos são os únicos que cobrem $\bar{w}xyz$, e $wxy\bar{z}$, respectivamente. Uma vez que esses dois produtos estejam incluídos, vemos que apenas um dos produtos restantes é necessário. Conseqüentemente, podemos tomar $\bar{w}z + wy\bar{z} + w\bar{x}y$ ou $\bar{w}z + wy\bar{z} + \bar{x}yz$ como a resposta final. ◀

Exercícios

1. **a)** Desenhe um mapa K para uma função em duas variáveis e ponha um 1 na célula que representa $\bar{x}y$.
b) Quais são os mintermos representados por células adjacentes a essa célula?

2. Encontre as expansões em soma de produtos representadas por estes mapas K.

a)

	y	$\bar{y}$
x	1	
$\bar{x}$	1	1

b)

	y	$\bar{y}$
x	1	1
$\bar{x}$		

c)

	y	$\bar{y}$
x	1	1
$\bar{x}$	1	1

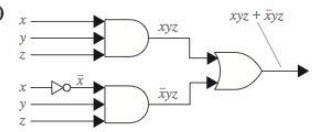
3. Desenhe os mapas K para estas expansões em soma de produtos em duas variáveis.
a) $x\bar{y}$ **b)** $xy + \bar{x}\bar{y}$
c) $xy + x\bar{y} + \bar{x}y + \bar{x}\bar{y}$

4. Use um mapa K para encontrar uma expansão mínima como uma soma booleana de produtos booleanos de cada uma destas funções das variáveis booleanas x e y.
a) $\bar{x}y + \bar{x}\bar{y}$
b) $xy + x\bar{y}$
c) $xy + x\bar{y} + \bar{x}y + \bar{x}\bar{y}$

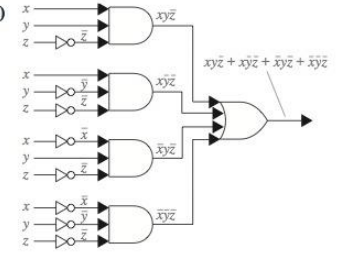
5. **a)** Desenhe um mapa K para uma função em três variáveis. Ponha um 1 na célula que representa $\bar{x}y\bar{z}$.
b) Quais mintermos são representados por células adjacentes a essa célula?

6. Use mapas K para encontrar circuitos mais simples com a mesma saída que cada um dos circuitos mostrados.

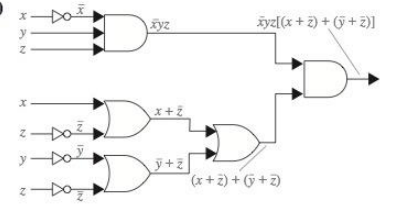
a)



b)



c)



7. Desenhe os mapas K destas expansões em soma de produtos em três variáveis.

a) $x\bar{y}\bar{z}$ **b)** $\bar{x}yz + \bar{x}\bar{y}\bar{z}$

c) $xyz + xy\bar{z} + \bar{x}y\bar{z} + \bar{x}\bar{y}z$

8. Construa um mapa K para $F(x, y, z) = xz + yz + xy\bar{z}$. Use esse mapa para encontrar implicantes, implicantes primos e implicantes primos essenciais de $F(x, y, z)$.

9. Construa um mapa K para $F(x, y, z) = x\bar{z} + xyz + y\bar{z}$. Use esse mapa para encontrar implicantes, implicantes primos e implicantes primos essenciais de $F(x, y, z)$.

10. Desenhe o 3-cubo Q_3 e rotule cada vértice com o mintermo nas variáveis booleanas x, y e z associado à seqüência de bits representada por este vértice. Para cada literal nestas variáveis, indique o 2-cubo Q_2 que é um subgrafo de Q_3 e representa este literal.

11. Desenhe o 4-cubo Q_4 e rotule cada vértice com o mintermo nas variáveis booleanas w, x, y e z associado à seqüência de bits representada por este vértice. Para cada literal nestas variáveis, indique o 3-cubo Q_3 que é um subgrafo de Q_4 e representa este literal. Indique qual 2-cubo Q_2, um subgrafo de Q_4, representa os produtos wz, $\bar{x}y$ e $\bar{y}\bar{z}$.

12. Use um mapa K para encontrar uma expansão mínima como uma soma booleana de produtos, para cada uma destas funções nas variáveis x, y e z.

a) $\bar{x}yz + x\bar{y}z$

b) $xyz + xy\bar{z} + \bar{x}yz + \bar{x}y\bar{z}$

c) $xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}z$

d) $xyz + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}y\bar{z} + \bar{x}\bar{y}\bar{z}$

13. **a)** Desenhe um mapa K para uma função em quatro variáveis. Ponha um 1 na célula que representa $\bar{w}xy\bar{z}$.

b) Quais mintermos são representados por células adjacentes a essa célula?

14. Use um mapa K para encontrar uma expansão mínima, como uma soma booleana de produtos, para cada uma destas funções nas variáveis w, x, y e z.

a) $wxyz + wx\bar{y}z + wx\bar{y}\bar{z} + w\bar{x}y\bar{z} + w\bar{x}\bar{y}z$

b) $wxy\bar{z} + wx\bar{y}z + w\bar{x}yz + \bar{w}xyz + \bar{w}x\bar{y}z + \bar{w}\bar{x}yz$

c) $wxyz + wxy\bar{z} + wx\bar{y}z + w\bar{x}\bar{y}z + w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}z + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z} + \bar{w}\bar{x}\bar{y}z$

d) $wxyz + wxy\bar{z} + wx\bar{y}z + w\bar{x}yz + w\bar{x}y\bar{z} + \bar{w}xyz + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z} + \bar{w}\bar{x}\bar{y}z$

15. Encontre as células em um mapa K para funções booleanas com cinco variáveis que correspondem a cada um destes produtos.

a) $x_1x_2x_3x_4$ **b)** $\bar{x}_1x_3x_5$ **c)** x_2x_4

d) $\bar{x}_3\bar{x}_4$ **e)** x_3 **f)** $\bar{x}_5$

16. Quantas células em um mapa K para funções booleanas com seis variáveis são necessárias para representar x_1, $\bar{x}_1x_6$, $\bar{x}_1x_2\bar{x}_6$, $x_2x_3x_4x_5$ e $x_1\bar{x}_2x_4\bar{x}_5$, respectivamente?

17. **a)** Quantas células tem um mapa K em seis variáveis?

b) Quantas células são adjacentes a determinada célula em um mapa K em seis variáveis?

18. Mostre que células em um mapa K para funções booleanas em cinco variáveis representam mintermos que diferem em exatamente um literal se e somente se eles são adjacentes ou estão em células que se tornam adjacentes quando as linhas de cima e as de baixo e as células na primeira e oitava colunas, na primeira e quarta colunas, na segunda e sétima colunas, na terceira e sexta colunas e na quinta e oitava colunas são consideradas adjacentes.

19. Quais linhas e quais colunas de um mapa 4×16 para funções booleanas em seis variáveis, usando os códigos de Gray 1111,1110,1010,1011,1001,1000,0000,0001,0011,0010, 0110,0111,0101,0100,1100,1101 para rotular as colunas e 11,10,00,01 para rotular as linhas, precisam ser consideradas adjacentes de modo que as células que representam mintermos que diferem em exatamente um literal sejam consideradas adjacentes?

***20.** Use mapas K para encontrar uma expansão mínima, como uma soma booleana de produtos booleanos de funções booleanas, que tem como entrada o código binário para cada algarismo decimal e produz como saída um 1 se e somente se o algarismo correspondente à entrada

a) for ímpar. **b)** não for divisível por três.

c) não for 4, 5 ou 6.

***21.** Suponha que existam cinco membros em um comitê, mas que Smith e Jones sempre votem o oposto de Marcus. Projete um circuito que implemente a votação majoritária no comitê usando esta relação entre os votos.

22. Use o método de Quine–McCluskey para simplificar as expansões em soma de produtos do Exemplo 3.

23. Use o método de Quine–McCluskey para simplificar as expansões em soma de produtos do Exercício 12.

24. Use o método de Quine–McCluskey para simplificar as expansões em soma de produtos do Exemplo 4.

25. Use o método de Quine–McCluskey para simplificar as expansões em soma de produtos do Exercício 14.

***26.** Explique como os mapas K podem ser usados para simplificar expansões em produto de somas em três variáveis. [*Dica:* Marque com um 0 todos os maxtermos em uma expansão e combine blocos de maxtermos.]

27. Use o método do Exercício 26 para simplificar a expansão em produto de somas $(x + y + z)(x + y + \bar{z})(x + \bar{y} + \bar{z})(x + \bar{y} + z)(\bar{x} + y + z)$.

***28.** Desenhe um mapa K para os 16 mintermos em quatro variáveis booleanas na superfície de um toro.

29. Construa um circuito usando portas *OU*, portas *E* e inversores que produza uma saída de 1 se um algarismo decimal, codificado usando uma expansão decimal codificada binária, for divisível por 3 e uma saída 0, caso contrário.

Nos exercícios 30 a 32, encontre uma expansão em soma de produtos mínima, dado o mapa K mostrado com condições *don't care* indicadas por *d*s.

30.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	*d*	1	*d*	1
$w\bar{x}$		*d*	*d*	
$\bar{w}\bar{x}$		*d*	1	
$\bar{w}x$		1	*d*	

31.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1			1
$w\bar{x}$		*d*	1	
$\bar{w}\bar{x}$		1	*d*	
$\bar{w}x$	*d*			*d*

32.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx		d	d	1
$w\bar{x}$	d	d	1	d
$\bar{w}\bar{x}$				
$\bar{w}x$	1	1	1	d

33. Mostre que os produtos de k literais correspondem a subcubos de dimensão 2^{n-k} do n-cubo Q_n, em que os vértices do cubo correspondem aos mintermos representados pelas seqüências de bits que rotulam os vértices, como descrito no Exemplo 8 da Seção 9.2.

Termos-chave e Resultados

TERMOS

variável booleana: uma variável que assume apenas os valores 0 e 1

$\bar{x}$ **(complemento de x):** uma expressão com o valor 1 quando x tiver o valor 0, e com o valor 0 quando x tiver o valor 1

$x \cdot y$ **(ou xy) (produto booleano** ou **conjunção de x e y):** uma expressão com o valor 1 quando ambos, x e y, tiverem o valor 1, e com o valor 0, caso contrário

$x + y$ **(soma booleana** ou **disjunção de x e y):** uma expressão com o valor 1 quando ou x ou y, ou ambos, tiverem o valor 1, e com o valor 0, caso contrário

expressões booleanas: expressões obtidas recursivamente para especificar que 0, 1, x_1, ..., x_n são expressões booleanas e $\bar{E}_1$, $(E_1 + E_2)$ e $(E_1 E_2)$ são expressões booleanas, se E_1 e E_2 o forem

dual de uma expressão booleana: expressão obtida trocando os sinais $+$ e os sinais $\cdot$ e trocando os 0s e os 1s

função booleana de n variáveis: uma função de B^n em B em que $B = \{0, 1\}$

álgebra booleana: um conjunto B com duas operações binárias $\vee$ e $\wedge$, elementos 0 e 1 e um operador de complementação $^-$ que satisfaz as propriedades dos elementos neutros, dos complementares, associativas, comutativas e distributivas

literal de uma variável booleana x: ou x ou $\bar{x}$

mintermo de $x_1, x_2, \ldots, x_n$: um produto booleano $y_1 y_2 \ldots y_n$, em que cada y_i é ou x_i ou $\bar{x}_i$

expansão em soma de produtos (ou **forma normal disjuntiva):** a representação de uma função booleana como uma disjunção de mintermos

funcionalmente completo: um conjunto de operadores booleanos é denominado funcionalmente completo se toda função booleana puder ser representada usando esses operadores

$x \mid y$ (ou ***x NE y***)**:** expressão que tem o valor 0 quando ambos, x e y, têm o valor 1, e tem o valor 1, caso contrário

$x \downarrow y$ (ou ***x NOU y***)**:** expressão que tem o valor 0 quando ou x ou y, ou ambos, têm o valor 1, e tem o valor 1, caso contrário

inversor: um aparelho que aceita o valor de uma variável booleana como entrada e produz o complemento da entrada

porta *OU*: um aparelho que aceita o valor de duas ou mais variáveis booleanas como entrada e produz sua soma booleana como saída

porta *E*: um aparelho que aceita os valores de duas ou mais variáveis booleanas como entrada e produz seu produto booleano como saída

meio somador: um circuito que soma dois bits, produzindo um bit soma e um bit "vai um"

somador completo: um circuito que soma dois bits e um "vai um", produzindo um bit soma e um bit "vai um"

mapa K para n variáveis: um retângulo dividido em 2^n células, em que cada célula representa um mintermo nas variáveis

minimização de uma função booleana: representação de uma função booleana como a soma do menor número de produtos de literais, tal que esses produtos contenham o menor número possível de literais entre todas as somas de produtos que representam esta função

implicante de uma função booleana: um produto de literais com a propriedade que, se este produto tiver o valor 1, então o valor desta função booleana é 1

implicante primo de uma função booleana: um produto de literais que é um implicante da função booleana e nenhum produto obtido excluindo um literal também é um implicante desta função

implicante primo essencial de uma função booleana: um implicante primo da função booleana que precisa ser incluído em uma minimização desta função

RESULTADOS

As identidades para álgebra booleana (veja a Tabela 5 na Seção 11.1).

Uma identidade entre funções booleanas representadas por expressões booleanas permanece válida quando são considerados os duais em ambos os lados da identidade.

Toda função booleana pode ser representada por uma expansão em soma de produtos.

Cada um dos conjuntos $\{+, ^-\}$ e $\{\cdot, ^-\}$ é funcionalmente completo.

Cada um dos conjuntos $\{\downarrow\}$ e $\{\mid\}$ é funcionalmente completo.

O uso dos mapas K para minimizar expressões booleanas.

O método de Quine–McCluskey para minimizar expressões booleanas.

Questões de Revisão

1. Defina uma função booleana de n variáveis.
2. Quantas funções booleanas de duas variáveis existem?
3. Dê uma definição recursiva do conjunto de expressões booleanas.
4. **a)** O que é o dual de uma expressão booleana?
   b) O que é o princípio de dualidade? Como ele pode ser usado para encontrar novas identidades que envolvem expressões booleanas?
5. Explique como construir uma expansão em soma de produtos de uma função booleana.
6. **a)** O que significa um conjunto de operadores ser funcionalmente completo?
   b) O conjunto $\{+, \cdot\}$ é funcionalmente completo?
   c) Existem conjuntos de um único operador que sejam funcionalmente completos?
7. Explique como construir um circuito para uma lâmpada controlada por dois interruptores usando portas *OU*, portas *E* e inversores.
8. Construa um meio somador usando portas *OU*, portas *E* e inversores.
9. Existe um tipo único de porta lógica que pode ser usada para construir todos os circuitos que podem ser construídos usando portas *OU*, portas *E* e inversores?
10. **a)** Explique como mapas K podem ser usados para simplificar expansões em soma de produtos em três variáveis booleanas.
    b) Use um mapa K para simplificar a expansão em soma de produtos $xyz + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}\bar{z}$.
11. **a)** Explique como mapas K podem ser usados para simplificar expansões em soma de produtos em quatro variáveis booleanas.
    b) Use um mapa K para simplificar a expansão em soma de produtos $wxyz + wx\bar{y}\bar{z} + w\bar{x}\bar{y}z + w\bar{x}\bar{y}\bar{z} + \bar{w}xyz + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}yz + \bar{w}\bar{x}\bar{y}\bar{z}$.
12. **a)** O que é uma condição *don't care*?
    b) Explique como as condições *don't care* podem ser usadas para construir um circuito com porta OU, portas E e inversores que produzam uma saída 1 se um algarismo decimal for maior que ou igual a 6 e uma saída 0 se esse algarismo for menor que 6.
13. **a)** Explique como usar o método de Quine–McCluskey para simplificar as expansões em soma de produtos.
    b) Use este método para simplificar $xy\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z} + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$.

Exercícios Complementares

1. Para quais valores das variáveis booleanas x, y e z vale que
   a) $x + y + z = xyz$?
   b) $x(y + z) = x + yz$?
   c) $\bar{x}\bar{y}\bar{z} = x + y + z$?
2. Sejam x e y pertencentes a $\{0, 1\}$. Segue necessariamente que $x = y$ se existir um valor z em $\{0, 1\}$, tal que
   a) $xz = yz$? **b)** $x + z = y + z$?
   c) $x \oplus z = y \oplus z$? **d)** $x \downarrow z = y \downarrow z$?
   e) $x \mid z = y \mid z$?

   Uma função booleana F é denominada **auto-dual** se e somente se $F(x_1, \ldots, x_n) = \overline{F}(\bar{x}_1, \ldots, \bar{x}_n)$.
3. Quais das seguintes funções são auto-duais?
   a) $F(x, y) = x$ **b)** $F(x, y) = xy + \bar{x}\bar{y}$
   c) $F(x, y) = x + y$ **d)** $F(x, y) = xy + \bar{x}y$
4. Dê um exemplo de uma função booleana auto-dual de três variáveis.

*5. Quantas funções booleanas de n variáveis são auto-duais?

Definimos a relação $\leq$ no conjunto das funções booleanas de n variáveis de modo que $F \leq G$, em que F e G são funções booleanas se e somente se $G(x_1, x_2, \ldots, x_n) = 1$ quando $F(x_1, x_2, \ldots, x_n) = 1$.

6. Determine se $F \leq G$ ou $G \leq F$ para os seguintes pares de funções.
   a) $F(x, y) = x$, $G(x, y) = x + y$
   b) $F(x, y) = x + y$, $G(x, y) = xy$
   c) $F(x, y) = \bar{x}$, $G(x, y) = x + y$
7. Mostre que, se F e G forem funções booleanas de n variáveis, então
   a) $F \leq F + G$. **b)** $FG \leq F$.
8. Mostre que, se F, G e H forem funções booleanas de n variáveis, então $F + G \leq H$ se e somente se $F \leq H$ e $G \leq H$.

*9. Mostre que a relação $\leq$ é uma ordem parcial no conjunto das funções booleanas de n variáveis.

*10. Desenhe o diagrama de Hasse para o poset que consiste no conjunto das 16 funções booleanas de duas variáveis (mostrado na Tabela 3 da Seção 11.1) com a ordem parcial $\leq$.

*11. Para cada uma destas igualdades, demonstre que ela é uma identidade ou encontre um conjunto de valores das variáveis para a qual ela não seja válida.
   a) $x \mid (y \mid z) = (x \mid y) \mid z$
   b) $x \downarrow (y \downarrow z) = (x \downarrow y) \downarrow (x \downarrow z)$
   c) $x \downarrow (y \mid z) = (x \downarrow y) \mid (x \downarrow z)$

Defina o operador booleano $\odot$ como segue: $1 \odot 1 = 1$, $1 \odot 0 = 0$, $0 \odot 1 = 0$ e $0 \odot 0 = 1$.

12. Mostre que $x \odot y = xy + \bar{x}\bar{y}$.
13. Mostre que $x \odot y = \overline{(x \oplus y)}$.
14. Mostre que cada uma destas identidades é válida.
   a) $x \odot x = 1$ **b)** $x \odot \bar{x} = 0$
   c) $x \odot y = y \odot x$
15. É sempre verdade que $(x \odot y) \odot z = x \odot (y \odot z)$?

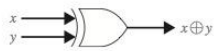
*16. Determine se o conjunto $\{\odot\}$ é funcionalmente independente.

*17. Quantas das 16 funções booleanas em duas variáveis x e y

podem ser representadas usando apenas o conjunto dado de operadores, as variáveis x e y e os valores 0 e 1?

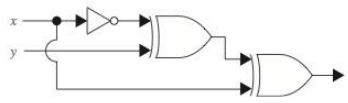
a) $\{^-\}$ **b)** $\{\cdot\}$ **c)** $\{+\}$ **d)** $\{\cdot, +\}$

A notação para uma **porta *XOU***, que produz a saída $x \oplus y$ de x e y, é a seguinte:

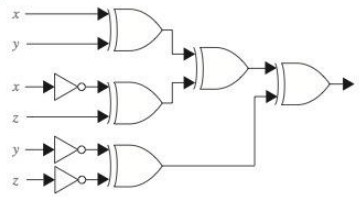


18. Determine a saída de cada um destes circuitos.

a)



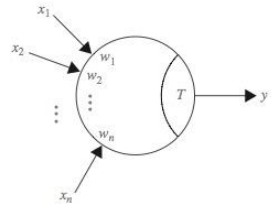
b)



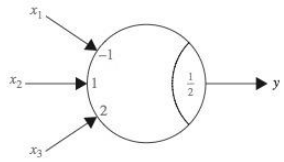
19. Mostre como um meio somador pode ser construído usando menos portas do que são usadas na Figura 8 da Seção 11.3 quando as portas *XOU* puderem ser usadas além das portas *OU*, portas *E* e inversores.

20. Projete um circuito que determina se três ou mais de quatro indivíduos em um comitê votam sim em uma questão, na qual cada indivíduo usa um interruptor para votar.

Uma **porta *threshold*** (porta limitadora) produz uma saída y que é 0 ou 1 dado um conjunto de valores de entrada para as variáveis booleanas $x_1, x_2, \ldots, x_n$. Uma porta *threshold* tem um **valor limiar** T, que é um número real, e **pesos** $w_1, w_2, \ldots, w_n$, cada um dos quais é um número real. A saída y da porta *threshold* é 1 se e somente se $w_1x_1 + w_2x_2 + \ldots + w_nx_n \geq T$. A porta *threshold* com valor limiar T e pesos $w_1, w_2, \ldots, w_n$ é representada pelo seguinte diagrama. Portas *threshold* são úteis na modelagem de neurofisiologia e em inteligência artificial.



21. Uma porta *threshold* representa uma função booleana. Encontre uma expressão booleana para a função booleana representada por esta porta *threshold*.



22. Uma função booleana que pode ser representada por uma porta *threshold* é chamada de **função *threshold***. Mostre que cada uma destas funções é uma função *threshold*.

a) $F(x) = \bar{x}$ **b)** $F(x, y) = x + y$

c) $F(x, y) = xy$ **d)** $F(x, y) = x \mid y$

e) $F(x, y) = x \downarrow y$ **f)** $F(x, y, z) = x + yz$

g) $F(w, x, y, z) = w + xy + z$

h) $F(w, x, y, z) = wxz + x\bar{y}z$

***23.** Mostre que $F(x, y) = x \oplus y$ não é uma função *threshold*.

***24.** Mostre que $F(w, x, y, z) = wx + yz$ não é uma função *threshold*.

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dados os valores de duas variáveis booleanas x e y, encontre os valores de $x + y$, xy, $x \oplus y$, $x \mid y$ e $x \downarrow y$.

2. Construa uma tabela que liste o conjunto de valores de todas as 256 funções booleanas de três variáveis.

3. Dados os valores de uma função booleana de n variáveis, em que n é um inteiro positivo, construa a expansão em soma de produtos desta função.

4. Dada a tabela de valores de uma função booleana, expresse essa função usando apenas os operadores $\cdot$ e $^-$.

5. Dada a tabela de valores de uma função booleana, expresse essa função usando apenas os operadores $+$ e $^-$.

***6.** Dada a tabela de valores de uma função booleana, expresse essa função usando apenas o operador $\mid$.

***7.** Dada a tabela de valores de uma função booleana, expresse essa função usando apenas o operador $\downarrow$.

8. Dada a tabela de valores de uma função booleana de três variáveis, construa seu mapa K.

9. Dada a tabela de valores de uma função booleana de quatro variáveis, construa seu mapa K.
**10. Dada a tabela de valores de uma função booleana, use o método de Quine–McCluskey para encontrar uma representação mínima em soma de produtos dessa função.
11. Dado um valor limiar e um conjunto de pesos para uma porta *threshold* e o valor de *n* variáveis booleanas na entrada, determine a saída dessa porta.
12. Dado um inteiro positivo *n*, construa uma expressão booleana aleatória em *n* variáveis na forma normal disjuntiva.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Calcule o número de funções booleanas com sete, oito, nove e dez variáveis.
2. Construa uma tabela de funções booleanas de três variáveis.
3. Construa uma tabela de funções booleanas de quatro variáveis.
4. Expresse cada uma das diferentes expressões booleanas em três variáveis na forma normal disjuntiva apenas com o operador *NE*, usando tão poucos operadores *NE* quanto possível. Qual é o maior número de operadores *NE* necessário?
5. Expresse cada uma das expressões booleanas diferentes em quatro variáveis na forma normal disjuntiva apenas com o operador *NOU*, usando tão poucos operadores *NOU* quanto possível. Qual é o maior número de operadores *NOU* necessário?
6. Gere aleatoriamente dez expressões booleanas diferentes em quatro variáveis e determine o número médio de passos necessários para minimizá-las usando o método Quine–McCluskey.
7. Gere aleatoriamente dez expressões booleanas diferentes em cinco variáveis e determine o número médio de passos necessários para minimizá-las usando o método Quine–McCluskey.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Descreva algumas das primeiras máquinas desenvolvidas para resolver problemas em lógica, tais como o Stanhope Demonstrator, Jevon's Logic Machine e Marquand Machine.
2. Explique a diferença entre circuitos combinatórios e circuitos seqüenciais. A seguir, explique como os *flip-flops* são usados para construir circuitos seqüenciais.
3. Defina um registrador *shift* e discuta como eles são usados. Mostre como construir registradores *shift* usando *flip-flops* e portas lógicas.
4. Mostre como podem ser construídos *multiplicadores* usando portas lógicas.
5. Descubra como as portas lógicas são fisicamente construídas. Discuta se as portas *NE* e *NOU* são usadas na construção de circuitos.
6. Explique como a *notação de dependência* pode ser usada para descrever circuitos de chaves complicados.
7. Descubra como os *multiplexers* são usados para construir circuitos de chaves.
8. Explique as vantagens de usar portas *threshold* para construir circuitos de chaves. Ilustre isso usando portas *threshold* para construir meios somadores e somadores completos.
9. Descreva o conceito de *circuitos de chaves hazard-free* e dê alguns dos princípios usados no projeto desses circuitos.
10. Explique como usar mapas K para minimizar funções de seis variáveis.
11. Discuta as idéias usadas pelos métodos mais novos para minimizar funções booleanas, tal como o Espresso. Explique como esses métodos podem ajudar a resolver problemas de minimização em até 25 variáveis.
12. Descreva o que se quer dizer com *decomposição funcional* de uma função booleana de *n* variáveis e discuta procedimentos para decompor funções booleanas em uma composição de funções booleanas com menos variáveis.

CAPÍTULO

12 Modelagem Computacional

12.1 Linguagens e Gramáticas
12.2 Máquinas de Estados Finitos com Saída
12.3 Máquinas de Estados Finitos sem Saída
12.4 Reconhecimento de Linguagem
12.5 Máquinas de Turing

Os computadores podem executar diversas tarefas. Dada uma tarefa, surgem duas questões. A primeira é: ela pode ser executada usando um computador? Uma vez que saibamos que essa primeira questão tem resposta afirmativa, podemos fazer a segunda questão: como a tarefa pode ser executada? Modelos computacionais são usados para responder a essas questões.

Vamos estudar três tipos de estruturas usadas em modelagem computacional, a saber, gramáticas, máquinas de estados finitos e máquinas de Turing. As gramáticas são usadas para gerar as palavras de uma linguagem e para determinar se uma palavra está em uma linguagem. As linguagens formais, que são geradas por gramáticas, fornecem modelos tanto para as linguagens naturais, tal como o português, como para as linguagens de programação, como Pascal, Fortran, Prolog, C e Java. Em particular, as gramáticas são extremamente importantes na construção e na teoria de compiladores. As gramáticas que discutiremos foram usadas inicialmente pelo lingüista americano Noan Chomsky na década de 1950.

Vários tipos de máquinas de estados finitos são usados em modelagem. Todas as máquinas de estados finitos têm um conjunto de estados, incluindo um estado inicial, um alfabeto de entrada e uma função de transição que associa o próximo estado a todo par de um estado e uma entrada. Os estados de uma máquina de estados finitos dão a ela recursos de memória limitados. Algumas máquinas de estados finitos produzem um símbolo de saída para cada transição; essas máquinas podem ser usadas para modelar muitos tipos de máquinas, incluindo máquinas de vendas, máquinas de tempo, somadores binários e reconhecedores de linguagem. Estudaremos também máquinas de estados finitos que não têm saída, mas possuem estados finais. Essas máquinas são amplamente usadas em reconhecimento de linguagem. As seqüências que são reconhecidas são aquelas que levam o estado inicial a um estado final. Os conceitos de gramáticas e máquinas de estados finitos podem ser ligados. Caracterizaremos aqueles conjuntos que são reconhecidos por uma máquina de estados finitos e mostraremos que estes são precisamente os conjuntos que são gerados por determinado tipo de gramática.

Finalmente, introduziremos o conceito de máquina de Turing. Mostraremos como essas máquinas podem ser usadas para reconhecer conjuntos. Mostraremos também como elas podem ser usadas para calcular funções *number-theoretic* (teoria dos números). Discutiremos a tese de Church-Turing, que afirma que todo cálculo efetivo pode ser executado usando uma máquina de Turing. Explicaremos como essas máquinas podem ser usadas para estudar as dificuldades na resolução de determinadas classes de problemas. Em particular, descreveremos como elas podem ser usadas para classificar problemas como tratáveis *versus* intratáveis, e solúveis *versus* insolúveis.

12.1 Linguagens e Gramáticas

Introdução

As palavras em português podem ser combinadas de várias formas. A gramática da língua portuguesa nos diz se uma combinação de palavras é uma sentença válida. Por exemplo, *o sapo escreve bem* é uma sentença válida, porque é formada por um predicado nominal, *o sapo*, composto pelo artigo *o* e pelo substantivo *sapo,* seguido por um predicado verbal, *escreve bem*, formado pelo verbo *escreve* e pelo advérbio *bem*. Não nos importa que esta seja uma afirmação sem sentido, pois estamos preocupados apenas com a **sintaxe**, ou forma, da sentença, e não com a **semântica**, ou significado. Observamos também que a combinação *nada rapidamente matemática* não é uma sentença válida, porque não segue as regras de gramática da língua portuguesa.

A sintaxe de uma **linguagem natural**, isto é, de uma linguagem falada, tal como português, francês, inglês, alemão ou espanhol, é extremamente complicada. De fato, não parece possível

especificar todas as regras de sintaxe para uma linguagem natural. Pesquisas na tradução automática de uma linguagem para outra levaram ao conceito de uma **linguagem formal**, a qual, diferentemente de uma linguagem natural, é especificada por um conjunto bem definido de regras de sintaxe. Regras de sintaxe são importantes não só em lingüística, o estudo das linguagens naturais, mas também no estudo das linguagens de programação.

Descreveremos as sentenças de uma linguagem formal usando uma gramática. O uso de gramáticas ajuda quando consideramos as duas classes de problemas que aparecem mais freqüentemente nas aplicações da linguagem de programação: (1) Como podemos determinar se uma combinação de palavras é uma sentença válida em uma linguagem formal? (2) Como podemos gerar as sentenças válidas de uma linguagem formal?

Antes de dar uma definição técnica de uma gramática, descreveremos um exemplo de uma gramática que gera um subconjunto da língua portuguesa. Esse subconjunto é definido usando uma lista de regras que descreve como uma sentença válida pode ser produzida. Especificamos que,

1. uma **sentença** é formada por um **predicado nominal** seguido de um **predicado verbal**;
2. um **predicado nominal** é formado por um **artigo**, seguido por um **adjetivo**, seguido por um **substantivo**, ou
3. um **predicado nominal** é formado por um **artigo** seguido por um **substantivo**;
4. um **predicado verbal** é formado por um **verbo** seguido por um **advérbio**, ou
5. um **predicado verbal** é formado por um **verbo**;
6. um **artigo** é *um*, ou
7. um **artigo** é *o*;
8. um **adjetivo** é *grande*, ou
9. um **adjetivo** é *esfomeado*;
10. um **substantivo** é *coelho*, ou
11. um **substantivo** é *matemático*;
12. um **verbo** é *come*, ou
13. um **verbo** é *pula*;
14. um **advérbio** é *rapidamente*, ou
15. um **advérbio** é *bastante*.

Dessas regras podemos formar sentenças válidas usando uma série de substituições até que nenhuma outra regra possa ser aplicada. Por exemplo, seguimos a seqüência de substituições:

sentença
predicado nominal **predicado verbal**
artigo **adjetivo** **substantivo** **predicado verbal**
artigo **adjetivo** **substantivo** **verbo** **advérbio**
o **adjetivo** **substantivo** **verbo** **advérbio**
o grande **substantivo** **verbo** **advérbio**
o grande coelho **verbo** **advérbio**
o grande coelho pula **advérbio**
o grande coelho pula rapidamente

para obter uma sentença válida. Também é fácil notar que algumas outras sentenças válidas são: *um matemático esfomeado come bastante*, *um matemático grande pula*, *o coelho come rapidamente*, e assim por diante. Além disso, podemos ver que *o rapidamente come matemático* não é uma sentença válida.

Gramáticas de Estrutura de Frase

Antes de darmos uma definição formal de uma gramática, introduziremos um pouco de terminologia.

DEFINIÇÃO 1 Um *vocabulário* (ou *alfabeto*) V é um conjunto finito, não vazio de elementos chamados de *símbolos*. Uma *palavra* (ou *sentença*) em V é uma seqüência de comprimento finito de elementos de V. A *seqüência vazia* ou *seqüência nula*, indicada por λ, é a seqüência que não contém nenhum símbolo. O conjunto de todas as palavras em V é indicado por V^*. Uma *linguagem em* V é um subconjunto de V^*.

Observe que λ, a seqüência vazia, é a seqüência que não contém nenhum símbolo. Ela é diferente de $\emptyset$, o conjunto vazio. Segue que $\{\lambda\}$ é o conjunto que contém exatamente uma seqüência, a saber, a seqüência vazia.

As linguagens podem ser especificadas de diversas maneiras. Uma delas é listar todas as palavras na linguagem. Outra é dar alguns critérios que uma palavra deve satisfazer para estar na linguagem. Nesta seção, descreveremos outra maneira importante de especificar uma linguagem, isto é, através do uso de uma gramática, tal como o conjunto de regras que demos na introdução desta seção. Uma gramática fornece um conjunto de símbolos de vários tipos e um conjunto de regras para produzir palavras. Mais precisamente, uma gramática tem um **vocabulário** V, que é um conjunto de símbolos usados para construir membros da linguagem. Alguns dos elementos do vocabulário não podem ser substituídos por outros símbolos. Estes são chamados de **terminais**, e os outros membros do vocabulário, que podem ser substituídos por outros símbolos, são chamados de **não terminais**. Os conjuntos terminais e os não terminais são usualmente indicados por T e N, respectivamente. No exemplo dado na introdução da seção, o conjunto dos terminais é {*um*, *o*, *coelho*, *matemático*, *pula*, *come*, *rapidamente*, *bastante*}, e o conjunto dos não terminais é {**sentença, predicado nominal, predicado verbal, adjetivo, artigo, substantivo, verbo, advérbio**}. Existe um membro especial do vocabulário chamado de **símbolo inicial**, indicado por S, que é o elemento do vocabulário com o qual sempre começamos. No exemplo da introdução, o símbolo inicial é **sentença**. As regras que especificam quando podemos substituir uma seqüência de V^*, o conjunto de todas as seqüências de elementos do vocabulário, por outra seqüência são chamadas de **regras de produção** da gramática. Indicamos por $z_0 \rightarrow z_1$ a regra de produção que especifica que z_0 pode ser substituída por z_1 em uma seqüência. As regras de produção na gramática apresentada na introdução desta seção foram listadas. A primeira regra de produção, escrita usando esta notação, é **sentença → predicado nominal predicado verbal**. Resumimos com a definição que a seguir.

DEFINIÇÃO 2 Uma *gramática de estrutura de frase* $G = (V, T, S, P)$ consiste em um vocabulário V, um subconjunto T de V que consiste nos elementos terminais, um símbolo inicial S de V e um conjunto finito de regras de produção P. O conjunto $V - T$ é indicado por N. Os elementos de N são chamados de *símbolos não terminais*. Toda regra de produção em P deve conter pelo menos um não terminal em seu lado esquerdo.

EXEMPLO 1 Seja $G = (V, T, S, P)$, em que $V = \{a, b, A, B, S\}$, $T = \{a, b\}$, S é o símbolo inicial e $P = \{S \rightarrow ABa, A \rightarrow BB, B \rightarrow ab, AB \rightarrow b\}$. G é um exemplo de uma gramática de estrutura de frase. ◀

Estaremos interessados em palavras que possam ser geradas por regras de produção de uma gramática de estrutura de frase.

DEFINIÇÃO 3 Seja $G = (V, T, S, P)$ uma gramática de estrutura de frase. Sejam $w_0 = lz_0r$ (isto é, a concatenação de l, z_0 e r) e $w_1 = lz_1r$ e seqüências em V. Se $z_0 \rightarrow z_1$ for uma regra de produção de G, dizemos que w_1 é *diretamente derivável* de w_0 e escrevemos $w_0 \Rightarrow w_1$. Se $w_0, w_1, \ldots, w_n$ forem seqüências em V, tal que $w_0 \Rightarrow w_1, w_1 \Rightarrow w_2, \ldots, w_{n-1} \Rightarrow w_n$, então dizemos que w_n é *derivável de* w_0 e escrevemos $w_0 \overset{*}{\Rightarrow} w_n$. A seqüência de passos usados para obter w_n a partir de w_0 é chamada de *derivação*.

EXEMPLO 2 A seqüência *Aaba* é diretamente derivável de *ABa* na gramática do Exemplo 1, pois $B \rightarrow ab$ é uma regra de produção na gramática. A seqüência *abababa* é derivável de *ABa*, pois $ABa \Rightarrow Aaba \Rightarrow BBaba \Rightarrow Bababa \Rightarrow abababa$, usando as regras de produção $B \rightarrow ab$, $A \rightarrow$ BB, $B \rightarrow ab$ e $B \rightarrow ab$ sucessivamente. ◀

DEFINIÇÃO 4 Seja $G = (V, T, S, P)$ uma gramática de estrutura de frase. A *linguagem gerada por G* (ou a *linguagem de G*), indicada por $L(G)$, é o conjunto de todas as seqüências de terminais que são deriváveis do estado inicial S. Em outras palavras,

$$L(G) = \{w \in T^* \mid S \overset{*}{\Rightarrow} w\}.$$

Nos exemplos 3 e 4, veremos linguagens geradas por uma gramática de estrutura de frase.

EXEMPLO 3 Seja G a gramática com vocabulário $V = \{S, A, a, b\}$, conjunto de terminais $T = \{a, b\}$, símbolo inicial S e regras de produção $P = \{S \rightarrow aA, S \rightarrow b, A \rightarrow aa\}$. O que é $L(G)$, a linguagem dessa gramática?

Solução: A partir do estado inicial S, podemos derivar aA usando a regra de produção $S \rightarrow aA$. Podemos também usar a regra de produção $S \rightarrow b$ para derivar b. De aA, a regra de produção $A \rightarrow aa$ pode ser usada para derivar aaa. Nenhuma palavra adicional pode ser derivada, portanto, $L(G) = \{b, aaa\}$. ◀

EXEMPLO 4 Seja G a gramática com vocabulário $V = \{S, 0, 1\}$, conjunto de terminais $T = \{0, 1\}$, símbolo inicial S e regras de produção $P = \{S \rightarrow 11S, S \rightarrow 0\}$. O que é $L(G)$, a linguagem dessa gramática?

Solução: A partir de S, podemos derivar 0 usando $S \rightarrow 0$, ou $11S$ usando $S \rightarrow 11S$. A partir de $11S$, podemos derivar ou 110 ou $1111S$. A partir de $1111S$, podemos derivar 11110 e $111111S$. Em qualquer estágio de uma derivação podemos ou adicionar dois 1s no final da seqüência ou encerrar a derivação adicionando um 0 no final da seqüência. Concluímos que $L(G) = \{0, 110, 11110, 1111110, ...\}$, o conjunto de todas as seqüências que começam com um número par de 1s e terminam com um 0. Isso pode ser demonstrado usando um argumento indutivo que mostra que depois de n regras de produção terem sido usadas, as únicas seqüências de terminais geradas são aquelas que consistem de $n - 1$ ou menos concatenações de 11 seguido por 0. (Isso é deixado como exercício para o leitor). ◀

O problema de construir uma gramática que gere uma dada linguagem aparece com freqüência. Os exemplos 5, 6 e 7 descrevem problemas desse tipo.

EXEMPLO 5 Dê uma gramática de estrutura de frase que gere o conjunto $\{0^n1^n \mid n = 0, 1, 2, ...\}$.

Solução: Duas regras de produção podem ser usadas para gerar todas as seqüências que consistem em uma seqüência de 0s seguida por uma seqüência de mesmo número de 1s, incluindo a seqüência nula. A primeira constrói seqüências cada vez mais longas na linguagem, adicionando um 0 no início da seqüência e um 1 no final. A segunda regra de produção substitui S por uma seqüência vazia. A solução é a gramática $G = (V, T, S, P)$, em que $V = \{0, 1, S,\}$, $T = \{0, 1\}$ S é o símbolo inicial e as regras de produção são

$S \rightarrow 0S1$

$S \rightarrow \lambda$.

A verificação de que esta gramática gera o conjunto correto é deixada como exercício para o leitor. ◀

O Exemplo 5 envolve o conjunto de seqüências formadas por 0s seguidos por 1s, nas quais o número de 0s e 1s é o mesmo. O Exemplo 6 considera o conjunto das seqüências que consistem em 0s seguidos por 1s, nas quais o número de 0s e 1s pode ser diferente.

EXEMPLO 6 Encontre uma gramática de estrutura de frase que gere o conjunto $\{0^m 1^n \mid m \text{ e } n \text{ são inteiros não negativos}\}$.

Solução: Daremos duas gramáticas G_1 e G_2 que geram esse conjunto. Isso ilustrará que duas gramáticas podem gerar a mesma linguagem.

A gramática G_1 tem alfabeto $V = \{S, 0, 1\}$; terminais $T = \{0, 1\}$; e regras de produção $S \to 0S$, $S \to S1$ e $S \to \lambda$. G_1 gera o conjunto correto, pois o uso da primeira regra de produção m vezes coloca m 0s no início da seqüência e o uso da segunda regra de produção n vezes coloca n 1s no final da seqüência. Os detalhes dessa verificação são deixados para o leitor.

A gramática G_2 tem alfabeto $V = \{S, A, 0, 1\}$; terminais $T = \{0, 1\}$; e regras de produção $S \to 0S$, $S \to 1A$, $S \to 1$, $A \to 1A$, $A \to 1$, e $S \to \lambda$. Os detalhes de que esta gramática gera o conjunto correto são deixados como exercício para o leitor. ◀

Algumas vezes, um conjunto que é fácil de descrever somente pode ser gerado por uma gramática complicada. O Exemplo 7 ilustra isso.

EXEMPLO 7 Um gramática que gera o conjunto $\{0^n 1^n 2^n \mid n = 0, 1, 2, 3, \ldots\}$ é $G = (V, T, S, P)$ com $V = \{0, 1, 2, S, A, B, C\}$; $T = \{0, 1, 2\}$; estado inicial S; e regras de produção $S \to C$, $C \to 0CAB$, $S \to \lambda$, $BA \to AB$, $0A \to 01$, $1A \to 11$, $1B \to 12$ e $2B \to 22$. Deixamos como exercício para o leitor (Exercício 12, no final desta seção) mostrar que esta afirmação está correta. A gramática dada é o tipo mais simples de gramática que gera este conjunto, em um significado que será esclarecido mais adiante nesta seção. ◀

Tipos de Gramáticas de Estrutura de Frase

As gramáticas de estrutura de frase podem ser classificadas de acordo com os tipos de regra de produção que são permitidas. Descreveremos o esquema de classificação introduzido por Noam Chomsky. Na Seção 12.4, veremos que diferentes tipos de linguagem definidos nesse esquema correspondem às classes de linguagem que podem ser reconhecidas usando-se diferentes modelos de máquinas computacionais.

Uma gramática de **tipo 0** não tem restrições em suas regras de produção. Uma gramática de **tipo 1** pode ter regras de produção da forma $w_1 \to w_2$, em que $w_1 = lAr$ e $w_2 = lwr$, em que A é um símbolo não terminal, l e r são seqüências de zero ou mais símbolos terminais ou não terminais e w é uma seqüência não vazia de símbolos terminais ou não terminais. Ela também pode ter a regra de produção $S \to \lambda$, contanto que S não apareça no lado direito de nenhuma outra regra de produção. Uma gramática de **tipo 2** pode ter apenas regras de produção do tipo $w_1 \to w_2$, em que w_1 é um único símbolo que não é um símbolo terminal. Uma gramática de **tipo 3** pode ter apenas regras de produção da forma $w_1 \to w_2$ com $w_1 = A$ e $w_2 = aB$ ou $w_2 = a$, em que A e B são símbolos não terminais e a é um símbolo terminal; ou com $w_1 = S$ e $w_2 = \lambda$.

As gramáticas de tipo 2 são chamadas de **gramáticas livres de contexto**, pois um símbolo não terminal, que é o lado esquerdo de uma regra de produção, pode ser substituído em uma seqüência sempre que ele ocorrer, não importando o que mais estiver na seqüência. Uma linguagem gerada por uma gramática de tipo 2 é chamada de **linguagem livre de contexto**. Quando existe uma regra de produção da forma $lw_1r \to lw_2r$ (mas não da forma $w_1 \to w_2$), a gramática é chamada de tipo 1 ou **sensível ao contexto**, pois w_1 pode ser substituído por w_2 apenas quando estiver rodeado pelas seqüências l e r. Uma linguagem gerada por uma gramática do tipo 1 é chamada **linguagem sensível ao contexto**. As gramáticas do tipo 3 também são chamadas de **gramáticas regulares**. Uma linguagem gerada por uma gramática regular é denominada **regular**. Na Seção 13.4 trataremos da relação entre gramáticas regulares e máquinas de estados finitos.

Dos quatro tipos de gramática que definimos, as gramáticas sensíveis ao contexto têm a definição mais complicada. Às vezes, essas gramáticas são definidas de modo diferente. Uma regra de

produção da forma $w_1 \rightarrow w_2$ é denominada **não contratora** se o comprimento de w_1 for menor que ou igual ao comprimento de w_2. De acordo com nossa caracterização de linguagens sensíveis ao contexto, toda regra de produção em uma gramática de tipo 1, diferente da regra de produção $S \rightarrow \lambda$, se ela estiver presente, é não contratora. Segue que os comprimentos das seqüências em uma derivação em uma linguagem livre de contexto são não decrescentes, a menos que seja usada a regra de produção $S \rightarrow \lambda$. Isso significa que a única maneira de a seqüência vazia pertencer à linguagem gerada por uma gramática sensível ao contexto é que a regra de produção $S \rightarrow \lambda$ seja parte da gramática. A outra maneira pela qual as gramáticas sensíveis ao contexto são definidas é especificando que todas as regras de produção são não contratoras. Uma gramática com essa propriedade é chamada de **não contratora** ou **monótona**. A classe das gramáticas não contratoras não é a mesma que a classe das gramáticas sensíveis ao contexto. Entretanto, essas duas classes estão intimamente ligadas; pode ser mostrado que elas definem a mesma linguagem, exceto pelo fato de as gramáticas não contratoras não poderem gerar nenhuma linguagem que contenha a seqüência λ.

EXEMPLO 8 Do exemplo 6 sabemos que $\{0^m1^n \mid m, n = 0, 1, 2, ...\}$ é uma linguagem regular, pois ela pode ser gerada por uma gramática regular, ou seja, a gramática G_2 no Exemplo 6. ◀

Exemplos Extras

As gramáticas livres de contexto e as regulares desempenham um papel importante nas linguagens de programação. As gramáticas livres de contexto são usadas para definir a sintaxe de quase todas as linguagens de programação. Essas gramáticas são suficientemente fortes para definir uma grande variedade de linguagens. Além disso, algoritmos eficientes podem ser desenvolvidos para determinar se e como uma seqüência pode ser gerada. As gramáticas regulares são usadas para pesquisar determinados padrões em textos e na análise léxica, que é o processo de transformar uma sentença de entrada em uma sentença de símbolos para uso por um *parser*.

EXEMPLO 9 Segue do Exemplo 5 que $\{0^n1^n \mid n = 0, 1, 2, ...\}$ é uma linguagem livre de contexto, pois as regras de produção dessa gramática são $S \rightarrow 0S1$ e $S \rightarrow \lambda$. Entretanto, ela não é uma linguagem regular. Isso será mostrado na Seção 12.4. ◀

EXEMPLO 10 O conjunto $\{0^n1^n2^n \mid n = 0, 1, 2, ...\}$ é uma linguagem sensível ao contexto, pois ele pode ser gerado por uma gramática do tipo 1, como mostra o Exemplo 7, mas não por qualquer linguagem do tipo 2. (Isso é mostrado no Exercício 28, nos exercícios suplementares no final deste capítulo.) ◀

A Tabela 1 resume a terminologia usada para classificar as gramáticas de estrutura de frase.

Árvores de Derivação

Uma derivação na linguagem gerada por uma gramática livre de contexto pode ser representada graficamente usando uma árvore ordenada enraizada, chamada de **árvore de derivação** ou ***parse tree***. A raiz dessa árvore representa o símbolo inicial. Os vértices internos da árvore representam os símbolos não terminais que aparecem na derivação. As folhas da árvore representam os símbolos terminais que aparecem. Se a regra de produção $A \rightarrow w$ aparecer na derivação, em que w é uma palavra, o vértice que representa A tem como filhos os vértices que representam cada símbolo em w, ordenados da esquerda para a direita.

TABELA 1 Tipos de Gramática.

Tipo	*Restrições nas Regras de Produção* $w_1 \rightarrow w_2$
0	Sem restrições
1	$w_1 = lAr$ e $w_2 = lwr$, em que $A \in N$, $l, r, w \in (N \cup T)^*$ e $w \neq \lambda$; ou $w_1 = S$ e $w_2 = \lambda$, contanto que S não esteja no lado direito de uma outra regra de produção
2	$w_1 = A$, em que A é um símbolo não terminal
3	$w_1 = A$ e $w_2 = aB$ ou $w_2 = a$, em que $A \in N$, $B \in N$, e $a \in T$; ou $w_1 = S$ e $w_2 = \lambda$

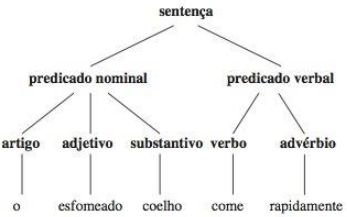


FIGURA 1 Árvore de Derivação.

EXEMPLO 11 Construa uma árvore de derivação para a derivação de *o esfomeado coelho come rapidamente*, dada na introdução desta seção.

Solução: A árvore de derivação é mostrada na Figura 1. ◀

O problema de determinar se uma seqüência está na linguagem gerada por uma gramática livre de contexto aparece em muitas aplicações, tal como a construção de compiladores. Duas abordagens desse problema são indicadas no Exemplo 12.

EXEMPLO 12 Determine se a palavra *cbab* pertence à linguagem gerada pela gramática $G = (V, T, S, P)$, em que $V = \{a, b, c, A, B, C, S\}$, $T = \{a, b, c\}$, S é o símbolo inicial e as regras de produção são

$S \rightarrow AB$
$A \rightarrow Ca$
$B \rightarrow Ba$
$B \rightarrow Cb$
$B \rightarrow b$
$C \rightarrow cb$
$C \rightarrow b$.

Solução: Uma maneira de abordar esse problema é começar com S e tentar derivar *cbab* usando uma série de regras de produção. Como existe apenas uma regra de produção com S no lado esquerdo, devemos começar com $S \Rightarrow AB$. A seguir, usamos a única regra de produção que tem A em seu lado esquerdo, a saber, $A \rightarrow Ca$, para obter $S \Rightarrow AB \Rightarrow CaB$. Como *cbab* começa com os símbolos *cb*, usamos a regra de produção $C \rightarrow cb$. Isso nos dá $S \Rightarrow Ab \Rightarrow CaB \Rightarrow cbaB$. Terminamos usando a regra de produção $B \rightarrow b$, para obter $S \Rightarrow AB \Rightarrow CaB \Rightarrow cbaB \Rightarrow cbab$. A abordagem que usamos é chamada de **top-down parsing**, porque ela começa com o símbolo inicial e continua pela aplicação sucessiva de regras de produção.

Existe uma outra abordagem para este problema, chamada de **bottom-up parsing**. Nessa abordagem, trabalhamos de trás para frente. Como *cbab* é a seqüência a ser derivada, podemos usar a regra de produção $C \rightarrow cb$, de modo que $Cab \Rightarrow cbab$. A seguir, podemos usar a regra de produção

Links

AVRAM NOAM CHOMSKY (NASCIDO EM 1928) Noam Chomsky, nascido na Filadélfia, é filho de um acadêmico hebreu. Ele obteve bacharelado, mestrado e doutorado em lingüística na Universidade da Pensilvânia. Esteve no corpo docente da Universidade da Pensilvânia de 1950 a 1951. Em 1955, foi contratado como professor no M.I.T. e começou sua carreira ensinando engenheiros franceses e alemães. Chomsky atualmente é professor de línguas estrangeiras e lingüística da cátedra Ferrari P. Ward, no M.I.T. Ele também é conhecido por suas diversas contribuições fundamentais à lingüística, incluindo o estudo de gramáticas. Chomsky também é amplamente conhecido por seu ativismo político.

$A \rightarrow Ca$, de modo que $Ab \Rightarrow Cab \Rightarrow cbab$. Usando a regra de produção $B \rightarrow b$, obtemos $AB \Rightarrow Ab \Rightarrow Cab \Rightarrow cbab$. Finalmente, usando $S \rightarrow AB$, temos que a derivação completa de *cbab* é $S \Rightarrow AB \Rightarrow Ab \Rightarrow Cab \Rightarrow cbab$. ◀

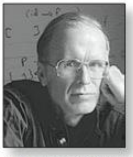
Forma de Backus–Naur

Links

Existe outra notação que algumas vezes é usada para especificar uma gramática do tipo 2, chamada de **forma de Backus–Naur (BNF)**, em homenagem a John Backus, que a inventou, e a Peter Naur, que a redefiniu para o uso na especificação da linguagem de programação ALGOL. (Surpreendentemente, uma notação bastante parecida com a forma de Backus–Naur foi usada há aproximadamente 2,5 mil anos para descrever a gramática do sânscrito.) A forma de Backus–Naur é usada para especificar as regras de sintaxe de muitas linguagens de computação, incluindo Java. As regras de produção de uma gramática do tipo 2 têm um único símbolo não terminal como seus lados esquerdos. Em vez de listar todas as regras de produção separadamente, podemos combinar todas com o mesmo símbolo não terminal do lado esquerdo em uma sentença. Em vez de usar o símbolo → em uma regra de produção, usamos o símbolo ::=. Colocamos todos os símbolos não terminais entre colchetes, ⟨ ⟩, e listamos todos os lados direitos das regras de produção na mesma sentença, separando-os por barras. Por exemplo, as regras de produção $A \rightarrow Aa$ $A \rightarrow a$ e $A \rightarrow AB$ podem ser combinadas em ⟨A⟩ ::= ⟨A⟩ a | a | ⟨A⟩ ⟨B⟩.

O Exemplo 13 ilustra como a forma de Backus–Naur é usada para descrever a sintaxe das linguagens de programação. Nosso exemplo vem da utilização original da forma de Backus–Naur na descrição do ALGOL 60.

Links



JOHN BACKUS (NASCIDO EM 1924) John Backus nasceu na Filadélfia e cresceu em Wilmington, Delaware. Freqüentou a Hill School em Pottstown, Pensilvânia. Ele precisou freqüentar a escola de verão todos os anos porque não gostava de estudar e não era um estudante muito sério. Entretanto, gostava de passar seus verões em New Hampshire, onde freqüentava a escola de verão e se divertia com as atividades, incluindo a vela. Para contentar seu pai, ele se inscreveu na universidade da Virgínia para estudar química. Contudo, decidiu rapidamente que química não era para ele, em 1943, entrou para as forças armadas, onde recebeu treinamento médico e trabalhou em uma ala de neurocirurgia em um hospital militar. Ironicamente, foi logo diagnosticado que Backus tinha um tumor ósseo em sua caixa craniana, que foi corrigida com uma placa de metal. Seu trabalho médico nas forças armadas o convenceu a tentar a escola de medicina, mas ele abandonou-a depois de nove meses, pois não gostou da "rotina de memorização" necessária. Depois de abandonar a escola de medicina, ingressou em uma escola para técnicos de rádio, pois queria construir seu próprio aparelho de alta-fidelidade. Nesta escola, um professor reconheceu seu potencial e pediu que ele o ajudasse com alguns cálculos matemáticos necessários em um artigo de uma revista. Finalmente, Backus encontrou aquilo por que se interessava: matemática e suas aplicações. Ele ingressou na universidade de Columbia, onde recebeu os títulos de bacharel e mestre em matemática. Backus ingressou na IBM, como programador, em 1950. Participou do projeto em desenvolvimento de dois dos primeiros computadores da IBM. De 1954 a 1958, liderou o grupo da IBM que desenvolveu o FORTRAN. Backus se tornou um membro permanente do pessoal da IBM Watson Research Center em 1958. Ele fez parte dos comitês que projetaram a linguagem de programação ALGOL, usando o que chamamos atualmente de forma de Backus–Naur para descrever a sintaxe dessa linguagem. Mais tarde, Backus trabalhou na matemática das famílias de conjuntos e em um estilo funcional de programação. Backus se tornou um pesquisador associado da IBM em 1963 e recebeu a National Medal of Science, em 1974, e o prestigioso Touring Award da Association of Computing Machinery em 1977.

PETER NAUR (NASCIDO EM 1928) Peter Naur nasceu em Frederiksberg, perto de Copenhagen. Quando menino, ele se interessou por astronomia. Não apenas observou os corpos celestes, mas também calculou a órbita de cometas e asteróides. Naur freqüentou a universidade de Copenhagen, tendo se formado em 1949. Ele passou 1950 e 1951 em Cambridge, onde usou um dos primeiros computadores para calcular o movimento de cometas e planetas. Depois de voltar para a Dinamarca, continuou trabalhando em astronomia, mas manteve seus laços com a computação. Em 1955, trabalhou como consultor na construção do primeiro computador dinamarquês. Em 1959, Naur mudou da astronomia para a computação como uma atividade em tempo integral. Seu primeiro trabalho como cientista da computação foi participar do desenvolvimento da linguagem de programação ALGOL. De 1960 a 1967, ele trabalhou no desenvolvimento de compiladores para ALGOL e COBOL. Em 1969, ele se tornou professor de ciência da computação na universidade de Copenhagen, onde trabalhou na área de metodologia de programação. Suas áreas de interesse em pesquisa incluem projeto, estrutura e desempenho de programas computacionais. Naur foi um pioneiro tanto na área de arquitetura de software quanto na de engenharia de software. Ele rejeita a visão de que a programação de computadores seja um ramo da matemática e prefere que a ciência da computação seja chamada de *datalogia*.

EXEMPLO 13 Em ALGOL 60, um identificador (que é o nome de uma entidade como uma variável) consiste em uma seqüência de caracteres alfanuméricos (isto é, letras ou algarismos) e deve começar com uma letra. Podemos usar estas regras no Backus–Naur para descrever o conjunto dos identificadores permitidos:

Exemplos Extras

⟨*identificador*⟩ ::= ⟨*letra*⟩ | ⟨*identificador*⟩ ⟨*letra*⟩ | ⟨*identificador*⟩ ⟨*algarismo*⟩
⟨*letra*⟩ ::= *a* | *b* |⋯ | *y* | *z* as reticências indicam que todas as 26 letras foram incluídas
⟨*algarismo*⟩ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Por exemplo, podemos produzir o identificador válido *x99a*, usando a primeira regra para substituir ⟨*identificador*⟩ por ⟨*identificador*⟩ ⟨*letra*⟩, a segunda regra para obter ⟨*identificador*⟩*a*, a primeira regra duas vezes para obter ⟨*identificador*⟩ ⟨*algarismo*⟩ ⟨*algarismo*⟩*a*, a terceira regra duas vezes para obter ⟨*identificador*⟩99*a*, a primeira regra para obter ⟨*letra*⟩99*a* e, finalmente, a segunda regra para obter *x99a*. ◀

EXEMPLO 14 Qual é a forma de Backus–Naur da gramática para o subconjunto da língua portuguesa descrito na introdução desta seção?

Solução: A forma de Backus–Naur dessa gramática é

⟨*sentença*⟩ ::= ⟨*predicado nominal*⟩ ⟨*predicado verbal*⟩
⟨*predicado nominal*⟩ ::= ⟨*artigo*⟩ ⟨*adjetivo*⟩ ⟨*substantivo*⟩ | ⟨*artigo*⟩ ⟨*substantivo*⟩
⟨*predicado verbal*⟩ ::= ⟨*verbo*⟩ ⟨*advérbio*⟩ | ⟨*verbo*⟩
⟨*artigo*⟩ ::= *um* | *o*
⟨*adjetivo*⟩ ::= *grande* | *esfomeado*
⟨*substantivo*⟩ ::= *coelho* | *matemático*
⟨*verbo*⟩ ::= *come* | *pula*
⟨*advérbio*⟩ ::= *rapidamente* | *bastante* ◀

EXEMPLO 15 Dê a forma de Backus–Naur para a produção de inteiros com sinais na notação decimal. (Um **inteiro com sinal** é um inteiro não negativo precedido por um sinal de mais ou por um sinal de menos.)

Solução: A forma de Backus–Naur para essa gramática que produz inteiros com sinal é

⟨*inteiro com sinal*⟩ ::= ⟨*sinal*⟩ ⟨*inteiro*⟩
⟨*sinal*⟩ ::= + | −
⟨*inteiro*⟩ ::= ⟨*algarismo*⟩ | ⟨*algarismo*⟩ ⟨*inteiro*⟩
⟨*algarismo*⟩ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 ◀

A forma de Backus–Naur com uma grande variedade de extensões é amplamente usada para especificar a sintaxe de linguagens de programação, tais como Java e LISP; linguagens de bancos de dados, tal como SQL; e *markup languages*, como o XML. Algumas extensões da forma de Backus–Naur comumente usadas na descrição das linguagens de programação são introduzidas no preâmbulo do Exercício 34 no final desta seção.

Exercícios

Os exercícios 1 a 3 referem-se à gramática com símbolo inicial **sentença**, conjunto de terminais $T = \{o, sonolento, alegre, tartaruga, lebre, passa, corre, rapidamente, lentamente\}$, conjunto de não terminais $N = \{$**predicado nominal, predicado verbal transitivo, predicado verbal intransitivo, artigo, adjetivo, substantivo, verbo, advérbio**$\}$ e regras de produção:

sentença → predicado nominal predicado verbal transitivo predicado nominal
sentença → predicado nominal predicado verbal intransitivo
predicado nominal → artigo adjetivo substantivo
predicado nominal → artigo substantivo
predicado verbal transitivo → verbo transitivo
predicado verbal intransitivo → verbo intransitivo advérbio
predicado verbal intransitivo → verbo intransitivo
artigo → *a*

adjetivo → *sonolento*
adjetivo → *alegre*
substantivo → *tartaruga*
substantivo → *lebre*
verbo transitivo → *passa*
verbo intransitivo → *corre*
advérbio → *rapidamente*
advérbio → *lentamente*

1. Use o conjunto de regras de produção para mostrar que cada uma destas sentenças é uma sentença válida.
a) *a alegre lebre corre*
b) *a sonolenta tartaruga corre rapidamente*
c) *a tartaruga passa a lebre*
d) *a sonolenta lebre passa a alegre tartaruga*

2. Encontre cinco outras sentenças válidas, além das apresentadas no Exercício 1.

3. Mostre que *a lebre corre a sonolenta tartaruga* não é uma sentença válida.

4. Seja $G = (V, T, S, P)$ a gramática de estrutura de frase com $V = \{0, 1, A, S\}$, $T = \{0, 1\}$ e o conjunto de regras de produção P com $S \to 1S$, $S \to 00A$, $A \to 0A$ e $A \to 0$.
a) Mostre que 111000 pertence à linguagem gerada por G.
b) Mostre que 11001 não pertence à linguagem gerada por G.
c) Qual é a linguagem gerada por G?

5. Seja $G = (V, T, S, P)$ a gramática de estrutura de frase com $V = \{0, 1, A, B, S\}$, $T = \{0, 1\}$ e o conjunto de regras de produção P com $S \to 0A$, $S \to 1A$, $A \to 0B$, $B \to 1A$, $B \to 1$.
a) Mostre que 10101 pertence à linguagem gerada por G.
b) Mostre que 10110 não pertence à linguagem gerada por G.
c) Qual é a linguagem gerada por G?

***6.** Sejam $V = \{S, A, B, a, b\}$ e $T = \{a, b\}$. Encontre a linguagem gerada pela gramática (V, T, S, P) quando o conjunto P de regras de produção consiste em
a) $S \to AB$, $A \to ab$, $B \to bb$.
b) $S \to AB$, $S \to aA$, $A \to a$, $B \to ba$.
c) $S \to AB$, $S \to AA$, $A \to aB$, $A \to ab$, $B, \to b$.
d) $S \to AA$, $S \to B$, $A \to aaA$, $A \to aa$, $B \to bB$, $B \to b$.
e) $S \to AB$, $A \to aAb$, $B \to bBa$, $A \to \lambda$., $B \to \lambda$.

7. Construa uma derivação de $0^3 1^3$ usando a gramática dada no Exemplo 5.

8. Mostre que a gramática dada no Exemplo 5 gera o conjunto $\{0^n 1^n \mid n = 0, 1, 2, ...\}$.

9. **a)** Construa uma derivação de $0^2 1^4$ usando a gramática G_1 do Exemplo 6.
b) Construa uma derivação de $0^2 1^4$ usando a gramática G_2 do Exemplo 6.

10. **a)** Mostre que a gramática G_1 dada no Exemplo 6 gera o conjunto $\{0^m 1^n \mid m, n = 0, 1, 2, \ldots\}$.
b) Mostre que a gramática G_2 do Exemplo 6 gera o mesmo conjunto.

11. Construa uma derivação de $0^2 1^2 2^2$ na gramática dada no Exemplo 7.

***12.** Mostre que a gramática dada no Exemplo 7 gera o conjunto $\{0^n 1^n 2^n \mid n = 0, 1, 2, \ldots\}$.

13. Encontre uma gramática de estrutura de frase para cada uma destas linguagens.
a) o conjunto que consiste nas seqüências de bits 0, 1 e 11
b) o conjunto das seqüências de bits que contém apenas 1s
c) o conjunto das seqüências de bits que começam com 0 e terminam com 1
d) o conjunto das seqüências de bits que consistem em um 0 seguido por um número par de 1s

14. Encontre uma gramática de estrutura de frase para cada uma destas linguagens.
a) o conjunto que consiste nas seqüências de bits 10, 01 e 101
b) o conjunto das seqüências de bits que começam com 00 e terminam com um ou mais 1s
c) o conjunto das seqüências de bits que consistem em um número par de 1s seguidos por um 0 final
d) o conjunto das seqüências de bits que não têm nem dois 0s consecutivos nem dois 1s consecutivos

***15.** Encontre uma gramática de estrutura de frase para cada uma destas linguagens.
a) o conjunto de todas as seqüências de bits que contêm um número par de 0s e nenhum 1
b) o conjunto de todas as seqüências de bits formadas por um 1 seguido por um número ímpar de 0s
c) o conjunto de todas as seqüências de bits que contenham um número par de 0s e um número par de 1s
d) o conjunto de todas as seqüências de bits que contêm dez ou mais 0s e nenhum 1
e) o conjunto de todas as seqüências de bits que contêm mais 0s do que 1s
f) o conjunto de todas as seqüências de bits que contêm o mesmo número de 0s e 1s
g) o conjunto de todas as seqüências de bits que contêm um número diferente de 0s e 1s

16. Construa gramáticas de estrutura de frase que gerem cada um destes conjuntos.
a) $\{1^n \mid n \geq 0\}$ **b)** $\{10^n \mid n \geq 0\}$
c) $\{(11)^n \mid n \geq 0\}$

17. Construa gramáticas de estrutura de frase que gerem cada um destes conjuntos.
a) $\{0^n \mid n \geq 0\}$ **b)** $\{1^n 0 \mid n \geq 0\}$
c) $\{(000)^n \mid n \geq 0\}$

18. Construa gramáticas de estrutura de frase que gerem cada um destes conjuntos.
a) $\{01^{2n} \mid n \geq 0\}$
b) $\{0^n 1^{2n} \mid n \geq 0\}$
c) $\{0^n 1^m 0^n \mid m \geq 0 \text{ e } n \geq 0\}$

19. Sejam $V = \{S, A, B, a, b\}$ e $T = \{a, b\}$. Determine se $G = \{V, T, S, P\}$ é uma gramática de tipo 0, mas não uma gramática de tipo 1, uma gramática de tipo 1, mas não uma gramática de tipo 2, ou uma gramática de tipo 2, mas não

uma gramática de tipo 3 se P, o conjunto das regras de produção, for

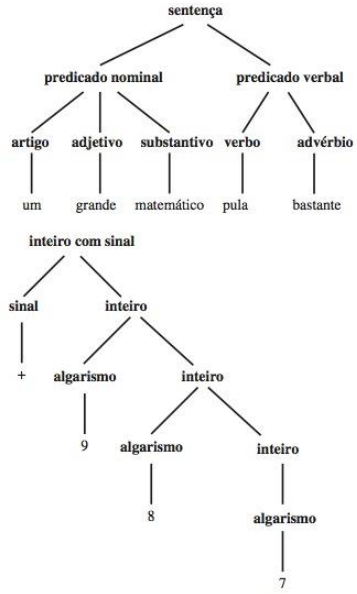
a) $S \to aAB, A \to Bb, B \to \lambda$.
b) $S \to aA, A \to a, A \to b$.
c) $S \to ABa, AB \to a$.
d) $S \to ABA, A \to aB, B \to ab$.
e) $S \to bA, A \to B, B \to a$.
f) $S \to aA, aA \to B, B \to aA, A \to b$.
g) $S \to bA, A \to b, S \to \lambda$.
h) $S \to AB, B \to aAb, aAb \to b$.
i) $S \to aA, A \to bB, B \to b, B \to \lambda$.
j) $S \to A, A \to B, B \to \lambda$.

20. Um **palíndromo** é uma seqüência que é lida do mesmo modo tanto de trás para a frente como no sentido usual, ou seja, uma seqüência w, em que $w = w^R$, em que w^R é o reverso da seqüência w. Encontre uma gramática livre de contexto que gere o conjunto de todos os palíndromos no alfabeto $\{0, 1\}$.

***21.** Sejam G_1 e G_2 gramáticas livres de contexto, que geram as linguagens $L(G_1)$ e $L(G_2)$, respectivamente. Mostre que existe uma gramática livre de contexto que gera cada um destes conjuntos.

a) $L(G_1) \cup L(G_2)$ **b)** $L(G_1)\,L(G_2)$
c) $L(G_1)$*

22. Encontre as seqüências construídas usando as árvores de derivação mostradas aqui.



23. Construa árvores de derivação para as sentenças do Exercício 1.

24. Seja G uma gramática com $V = \{a, b, c, S\}$; $T = \{a, b, c\}$; símbolo inicial S; e regras de produção $S \to abS$, $S \to bcS$, $S \to bbS$, $S \to a$ e $S \to cb$. Construa árvores de derivação para

a) *bcbba*.
b) *bbbcbba*.
c) *bcabbbbbcb*.

***25.** Use top-down parsing para determinar se cada uma das seguintes seqüências pertence à linguagem gerada pela gramática do Exemplo 12.

a) *baba* **b)** *abab*
c) *cbaba* **d)** *bbbcba*

***26.** Use bottom-up parcing para determinar se as seqüências do Exercício 25 pertencem à linguagem gerada pela gramática do Exemplo 12.

27. Construa uma árvore de derivação para -109, usando a gramática dada no Exemplo 15.

28. **a)** Explique o que são as regras de produção em uma gramática, se a forma de Backus–Naur para as regras de produção for a seguinte:

⟨*expressão*⟩ ::= (⟨*expressão*⟩) |
⟨*expressão*⟩ + ⟨*expressão*⟩ |
⟨*expressão*⟩ ∗ ⟨*expressão*⟩ |
⟨*variável*⟩
⟨*variável*⟩ ::= $x \mid y$

b) Encontre uma árvore de derivação para $(x * y) + x$ nesta gramática.

29. **a)** Construa uma gramática de estrutura de frase que gere todos os números decimais com sinal, que consistem em um sinal, ou + ou −; um inteiro não negativo e uma fração decimal que é ou a seqüência vazia ou um ponto decimal seguido por um inteiro positivo, em que são permitidos zeros iniciais em um inteiro.
b) Dê a forma de Backus–Naur desta gramática.
c) Construa uma árvore de derivação para $-31{,}4$ nesta gramática.

30. **a)** Construa uma gramática de estrutura de frase para o conjunto de todas as frações da forma $a \,/\, b$, em que a é um inteiro com sinal na notação decimal e b é um inteiro positivo.
b) Qual é a forma de Backus–Naur para esta gramática?
c) Construa uma árvore de derivação para $+311/17$ nesta gramática.

31. Dê as regras de produção na forma de Backus–Naur para um identificador se ele puder consistir em

a) uma ou mais letras minúsculas.
b) pelo menos três, mas não mais de seis letras minúsculas.
c) de uma a seis letras maiúsculas ou minúsculas, começando com uma letra maiúscula.
d) Uma letra minúscula, seguida por um algarismo ou um *sublinhado*, seguido por três ou quatro caracteres alfanuméricos (letras maiúsculas ou minúsculas e algarismos).

32. Dê as regras de produção na forma de Backus–Naur para o nome de uma pessoa se este nome consistir em um prenome, que é uma seqüência de letras, na qual apenas a primeira letra é maiúscula; uma inicial no meio; e um sobrenome, que pode ser qualquer seqüência de letras.

33. Dê regras de produção na forma de Backus–Naur que gerem todos os identificadores na linguagem de programação C.

Em C, um identificador começa com uma letra ou um *sublinhado* (_) que é seguido por uma ou mais letras minúsculas, letras maiúsculas, *underscores* e algarismos.

Diversas extensões da forma de Backus–Naur são comumente usadas para definir gramáticas de estrutura de frase. Em uma dessas extensões, um ponto de interrogação (?) indica que o símbolo, ou grupo de símbolos dentro dos parênteses, à sua esquerda pode aparecer zero ou uma vez (isto é, ele é opcional), um asterisco (*) indica que o símbolo à sua esquerda pode aparecer zero ou mais vezes e um sinal de mais (+) indica que o símbolo à sua esquerda pode aparecer uma ou mais vezes. Essas extensões são partes da **forma de Backus–Naur estendida (EBNF),** e os símbolos ?, * e + são chamados de **metacaracteres**. Em EBNF, os colchetes usados para indicar não terminais usualmente não são mostrados.

34. Descreva o conjunto de seqüências definidas por estes conjuntos de regras de produção em EBNF.

a) *seqüência* ::= $L+D?L+$
$L ::= a \mid b \mid c$
$D ::= 0 \mid 1$

b) *seqüência* ::= *sinal* $D+ \mid D+$
sinal ::= $+ \mid -$
$D ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

c) *seqüência* ::= $L*(D+)?L*$
$L ::= x \mid y$
$D ::= 0 \mid 1$

35. Dê regras de produção na forma de Backus–Naur estendida que gerem todos os números decimais que consistem em um sinal opcional, um inteiro não negativo e uma fração decimal que é ou a seqüência vazia ou um ponto decimal seguido por um inteiro positivo facultativo, opcionalmente precedido por determinado número de zeros.

36. Dê regras de produção na forma de Backus–Naur estendida que gerem um sanduíche, se um sanduíche consistir em uma fatia de pão embaixo, mostarda ou maionese; alface opcional; uma fatia opcional de tomate; uma ou mais fatias de peru, frango ou rosbife (em qualquer combinação); opcionalmente algum número de fatias de queijo; e uma fatia de pão em cima.

37. Dê regras de produção na forma de Backus–Naur estendida para identificadores na linguagem de programação C (veja o Exercício 33).

38. Descreva como as regras de produção para uma gramática na forma de Backus–Naur estendida podem ser traduzidas em um conjunto de regras de produção para a gramática na forma de Backus–Naur.

Esta é a forma de Backus–Naur que descreve a sintaxe de expressões na notação pós-fixa (ou polonesa reversa).

⟨*expressão*⟩ ::= ⟨*termo*⟩ | ⟨*termo*⟩ ⟨*termo*⟩ ⟨*operador de soma*⟩
⟨*operadora de soma*⟩ ::= $+ \mid -$
⟨*termo*⟩ ::= ⟨*fator*⟩ | ⟨*fator*⟩ ⟨*fator*⟩ ⟨*operador de multiplicação*⟩
⟨*operador de multiplicação*⟩ ::= $* \mid /$
⟨*fator*⟩ ::= ⟨*identificador*⟩ | ⟨*expressão*⟩
⟨*identificador*⟩ ::= $a \mid b \mid \cdots \mid z$

39. Para cada uma destas seqüências, determine se ela á gerada pela gramática dada para a notação pós-fixa. Se ela for, encontre os passos usados para gerar a seqüência

a) $abc*+$ **b)** $xy++$ **c)** $xz-z*$
d) $wxyz-*/$ **e)** $ade-*$

40. Use a forma de Backus–Naur para descrever a sintaxe de expressões na notação infixa, em que o conjunto de operadores e identificadores é o mesmo que na BNF para expressões pós-fixas dada no preâmbulo do Exercício 39, mas as expressões usadas como fatores devem ser rodeadas por parênteses.

41. Para cada uma destas seqüências, determine se ela é gerada pela gramática para expressões infixas do Exercício 40. Se ela for, encontre os passos usados para gerar a seqüência.

a) $x + y + z$ **b)** $a/b + c/d$
c) $m * (n + p)$ **d)** $+ m - n + p - q$
e) $(m + n) * (p - q)$

42. Seja G uma gramática e seja R a relação que contém os pares ordenados (w_0, w_1) se e somente se w_1 for diretamente derivável de w_0 em G. Qual é o fecho reflexivo transitivo de R?

12.2 Máquinas de Estados Finitos com Saída

Introdução

Links

Muitos tipos de máquinas, incluindo componentes em computadores, podem ser modelados usando uma estrutura chamada máquina de estados finitos. Diversos tipos de máquinas de estados finitos são comumente usados em modelos. Todas essas versões de máquinas de estados finitos incluem um conjunto finito de estados, com um estado inicial escolhido, um alfabeto de entrada e uma função de transição, que associa um estado seguinte a cada par estado e entrada. Máquinas de estados finitos são amplamente usadas em aplicações na ciência da computação e em rede de dados. Por exemplo, máquinas de estados finitos são a base para programas para verificação ortográfica, verificação gramatical, indexação ou busca em grandes textos, reconhecimento de fala, transformação de texto usando *markup languages*, como XML e HTML, e protocolos de redes que especificam como os computadores se comunicam.

TABELA 1 Tabela de Estados para uma Máquina de Vendas.

	Próximo Estado					Saída				
	Entrada					*Entrada*				
Estado	**5**	**10**	**25**	***O***	***R***	**5**	**10**	**25**	***O***	***R***
s_0	s_1	s_2	s_5	s_0	s_0	*n*	*n*	*n*	*n*	*n*
s_1	s_2	s_3	s_6	s_1	s_1	*n*	*n*	*n*	*n*	*n*
s_2	s_3	s_4	s_6	s_2	s_2	*n*	*n*	5	*n*	*n*
s_3	s_4	s_5	s_6	s_3	s_3	*n*	*n*	10	*n*	*n*
s_4	s_5	s_6	s_6	s_4	s_4	*n*	*n*	15	*n*	*n*
s_5	s_6	s_6	s_6	s_5	s_5	*n*	5	20	*n*	*n*
s_6	s_6	s_6	s_6	s_0	s_0	5	10	25	OJ	AJ

Nesta seção, estudaremos aquelas máquinas de estados finitos que produzem saídas. Mostraremos como as máquinas de estados finitos podem ser usadas para modelar máquinas de vendas, uma máquina que atrasa a entrada, uma máquina que soma inteiros e uma máquina que determina se uma seqüência de bits contém um padrão especificado.

Antes de dar definições formais, mostraremos como uma máquina de vendas pode ser modelada. Uma máquina de vendas aceita moedas de 5 centavos, 10 centavos e 25 centavos. Quando um total de 30 centavos ou mais for depositado, a máquina imediatamente devolve a quantia que passar de 30 centavos. Quando 30 centavos forem depositados e todo o excesso devolvido, o consumidor pode apertar um botão laranja e receber um suco de laranja, ou apertar um botão vermelho e receber um suco de maçã. Podemos descrever como a máquina funciona especificando seus estados, como ela muda de estado quando recebe uma entrada e a saída que é produzida para toda a combinação de entrada e estado corrente.

A máquina pode estar em qualquer um de sete estados diferentes s_i, $i = 0, 1, 2, \ldots, 6$, em que s_i é o estado no qual a máquina coletou $5i$ centavos. A máquina começa no estado s_0, com 0 centavos recebidos. As entradas possíveis são 5 centavos, 10 centavos, 25 centavos, o botão laranja (O) e o vermelho (R). As possíveis saídas são nada (n), 5 centavos, 10 centavos, 15 centavos, 20 centavos, 25 centavos, um suco de laranja e um suco de maçã.

Ilustramos como esse modelo da máquina funciona com este exemplo. Suponha que um estudante ponha uma moeda de 10 centavos seguida por uma de 25 centavos, receba 5 centavos de volta e então aperte o botão laranja para um suco de laranja. A máquina começa no estado s_0.

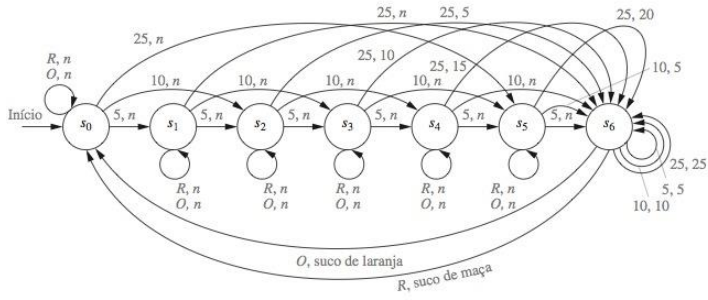


FIGURA 1 Máquina de Venda.

A primeira entrada é 10 centavos, que muda o estado da máquina para s_2 e não produz nenhuma saída. A segunda entrada é 25 centavos. Isso muda o estado de s_2 para s_6 e devolve 5 centavos como saída. A próxima entrada é o botão laranja, que muda o estado de s_6 de volta para s_0 (pois a máquina retorna ao seu estado inicial) e fornece um suco de laranja com sua saída.

Podemos mostrar todas as mudanças de estado e as saídas dessa máquina em uma tabela. Para fazer isso, precisamos especificar para cada combinação de estado e entrada o próximo estado e saída obtidos. A Tabela 1 mostra as transições e as saídas para cada par de um estado e uma entrada.

Outra maneira de mostrar as ações de uma máquina é usar um grafo orientado com arestas rotuladas, em que cada estado é representado por um círculo, as arestas representam as transições e estão rotuladas com a entrada e a saída para aquela transição. A Figura 1 mostra esse grafo orientado para a máquina de vendas.

Máquinas de Estados Finitos com Saídas

Daremos agora a definição formal de uma máquina de estados finitos com saída.

DEFINIÇÃO 1 Uma *máquina de estados finitos* $M = (S, I, O, f, g, s_0)$ consiste em um conjunto finito S de *estados*, um *alfabeto de entrada* finito I, um *alfabeto de saída* finito O, uma *função de transição* f que associa a cada par de estado e entrada um novo estado, uma *função de saída* g que associa a cada par de estado e entrada uma saída, e um estado inicial s_0.

Seja $M = (S, I, O, f, g, s_0)$ uma máquina de estados finitos. Podemos usar uma **tabela de estados** para representar os valores da função de transição f e da função de saída g para todos os pares de estado e entrada. Construímos anteriormente uma tabela de estados para a máquina de vendas examinada na introdução desta seção.

EXEMPLO 1 A tabela de estados mostrada na Tabela 2 descreve uma máquina de estados finitos com $S = \{s_0, s_1, s_2, s_3\}$, $I = \{0, 1\}$ e $O = \{0, 1\}$. Os valores da função de transição f são mostrados nas duas primeiras colunas e os valores da função de saída g são mostrados nas duas últimas colunas. ◀

Outra maneira de representar uma máquina de estados finitos é usar um **diagrama de estados**, que é um grafo orientado com arestas rotuladas. Nesse diagrama, cada estado é representado por um círculo. Flechas rotuladas com pares de entrada e saída são mostradas para cada transição.

EXEMPLO 2 Construa o diagrama de estados para a máquina de estados finitos com a tabela de estados mostrada na Tabela 2.

Solução: O diagrama de estados para essa máquina é mostrado na Figura 2. ◀

EXEMPLO 3 Construa a tabela de estados para a máquina de estados finitos com diagrama de estados mostrada na Figura 3.

Solução: A tabela de estados para essa máquina é mostrada na Tabela 3. ◀

Uma seqüência de entrada leva o estado inicial por uma seqüência de estados, como determinado pela função de transição. À medida que lemos a seqüência de entrada símbolo por símbolo (da esquerda para a direita), cada símbolo de entrada leva a máquina de um estado para outro. Como cada transição produz uma saída, uma seqüência de entrada produz também uma seqüência de saída.

Suponha que a seqüência de entrada seja $x = x_1x_2 \ldots x_k$. Então, a leitura dessa entrada leva a máquina do estado s_0 para o estado s_1, em que $s_1 = f(s_0, x_1)$, a seguir para o estado s_2, em que $s_2 =$

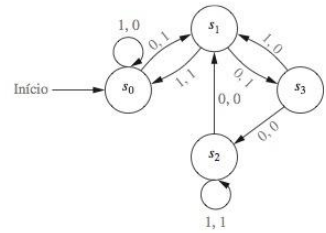


FIGURA 2 O Diagrama de Estado para a Máquina de Estados Finitos Exibida na Tabela 2.

TABELA 2				
	f		g	
	Entrada		*Entrada*	
Estado	**0**	**1**	**0**	**1**
s_0	s_1	s_0	1	0
s_1	s_3	s_0	1	1
s_2	s_1	s_2	0	1
s_3	s_2	s_1	0	0

$f(s_1, x_2)$, e assim por diante, com $s_j = f(s_{j-1}, x_j)$ para $j = 1, 2, \ldots, k$, terminando no estado $s_k = f(s_{k-1}, x_k)$. Esta seqüência de transições produz uma seqüência de saída $y_1y_2 \ldots y_k$, em que $y_1 = g(s_0, x_1)$ é a saída correspondente à transição de s_0 para s_1, $y_2 = g(s_1, x_2)$ é a saída correspondente à transição de s_1 para s_2, e assim por diante. Em geral, $y_j = g(s_{j-1}, x_j)$ para $j = 1, 2, \ldots, k$. Logo, podemos estender a definição da função de saída g para seqüências de entrada de modo que $g(x) = y$, em que y é a saída correspondente à seqüência de entrada x. Esta notação é útil em muitas aplicações.

EXEMPLO 4 Encontre a seqüência de saída gerada pela máquina de estados finitos na Figura 3, se a seqüência de entrada for 101011.

Solução: A saída obtida é 001000. Os estados e as saídas sucessivos são mostrados na Tabela 4. ◀

Podemos agora olhar para alguns exemplos de máquinas de estados finitos úteis. Os exemplos 5, 6 e 7 ilustram que os estados de uma máquina de estados finitos dá a ela recursos limitados de memória. Os estados podem ser usados para lembrar as propriedades dos símbolos que foram lidos pela máquina. Entretanto, como há apenas um número finito de estados diferentes, as máquinas de estados finitos não podem ser usadas para alguns propósitos importantes. Isso será ilustrado na Seção 12.4.

EXEMPLO 5 Um elemento importante em muitos aparelhos eletrônicos é a *máquina de atraso unitário* (*unit-delay machine*), que produz como saída a seqüência de entrada atrasada por uma quantidade de tempo especificada. Como pode ser construída uma máquina de estados finitos que atrase uma seqüência de entrada por uma unidade de tempo, ou seja, produza como saída a seqüência de bits $0x_1x_2 \ldots x_{k-1}$, dada a seqüência de bits de entrada $x_1x_2 \ldots x_k$?

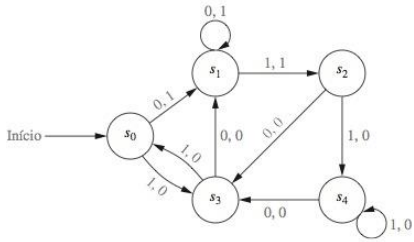


FIGURA 3 Máquina de Estados Finitos.

TABELA 3				
	f		g	
	Entrada		*Entrada*	
Estado	**0**	**1**	**0**	**1**
s_0	s_1	s_3	1	0
s_1	s_1	s_2	1	1
s_2	s_3	s_4	0	0
s_3	s_1	s_0	0	0
s_4	s_3	s_4	0	0

TABELA 4							
Entrada	1	0	1	0	1	1	—
Estado	s_0	s_3	s_1	s_2	s_3	s_0	s_3
Saída	0	0	1	0	0	0	—

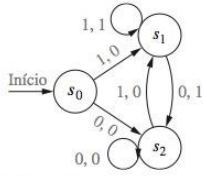


FIGURA 4 Máquina de Atraso Unitário.

Solução: Pode ser construída uma máquina de atraso que tenha duas entradas possíveis, a saber, 0 e 1. A máquina deve ter um estado inicial s_0. Como a máquina deve lembrar se a entrada anterior era um 0 ou um 1, dois outros estados s_1 e s_2 são necessários, em que a máquina está no estado s_1 se a entrada anterior foi 1 e no estado s_2, se a entrada anterior foi 0. Uma saída de 0 é produzida para a transição inicial a partir de s_0. Cada transição a partir de s_1 dá uma saída de 1 e cada transição a partir de s_2 dá uma saída de 0. A saída correspondente à seqüência de entrada $x_1 \ldots x_k$ é a seqüência que começa com 0, seguida por x_1, seguida por $x_2 \ldots$, terminando com x_{k-1}. O diagrama de estado para esta máquina é mostrado na Figura 4. ◀

EXEMPLO 6 Produza uma máquina de estados finitos que some dois inteiros usando suas expansões binárias.

Solução: Quando $(x_n \ldots x_1x_0)_2$ e $(y_n \ldots y_1y_0)_2$ são somados, o seguinte procedimento (como descrito na Seção 3.6) é seguido. Primeiro, os bits x_0 e y_0 são somados, produzindo um bit soma z_0 e um bit "vai um" c_0. Este bit "vai um" é ou 0 ou 1. A seguir, os bits x_1 e y_1 são somados juntamente com o bit "vai um" c_0. Isso dá um bit soma z_1 e um bit "vai um" c_1. Esse procedimento continua até o n-ésimo estágio, em que x_n, y_n e o "vai um" anterior c_{n-1} são somados para produzir o bit soma z_n e o bit "vai um" c_n, que é igual ao bit soma z_{n-1}.

Uma máquina de estados finitos para executar essa adição pode ser construída usando apenas dois estados. Por simplicidade, supomos que ambos os bits iniciais x_n e y_n sejam 0 (caso contrário, teríamos de tomar cuidados especiais com relação ao bit soma z_{n+1}). O estado inicial s_0 é usado para lembrar que o "vai um" anterior é 0 (ou para a adição dos bits mais à direita). O outro estado, s_1, é usado para lembrar que o "vai um" anterior é 1.

Como as entradas da máquina são pares de bits, existem quatro entradas possíveis. Representamos essas possibilidades por 00 (quando ambos os bits forem 0) , 01 (quando o primeiro bit for 0 e o segundo for 1), 10 (quando o primeiro bit for 1 e o segundo for 0), e 11 (quando ambos os bits forem 1). As transições e as saídas são construídas a partir da soma dos dois bits representados pela entrada e pelo "vai um" representado pelo estado. Por exemplo, quando a máquina está no estado s_1 e recebe 01 como entrada, o próximo estado é s_1 e a saída é 0, porque a soma que aparece é $0 + 1 + 1 = (10)_2$. O diagrama de estado para essa máquina é mostrado na Figura 5. ◀

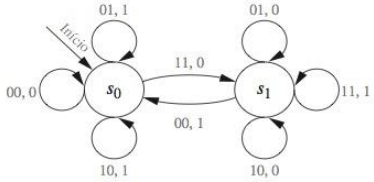


FIGURA 5 Máquina de Estados Finitos para a Soma.

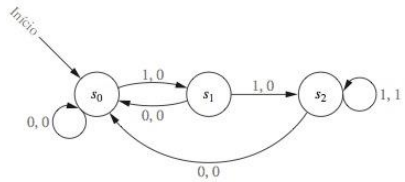


FIGURA 6 Máquina de Estados Finitos que Dá Uma Saída 1 se e somente se a Seqüência de Entrada Lida até o Momento Acaba com 111.

EXEMPLO 7 Em determinado esquema de codificação, quando três 1s consecutivos aparecem um uma mensagem, o receptor da mensagem sabe que houve um erro de transmissão. Construa uma máquina de estados finitos que dê um 1 como seu bit de saída corrente se e somente se os últimos três bits recebidos forem todos 1s.

Solução: São necessários três estados nesta máquina. O estado inicial s_0 lembra que o valor de entrada anterior, se existir, não era um 1. O estado s_1 lembra que a entrada anterior era um 1, mas a penúltima entrada, se existir, não era um 1. O estado s_2 lembra que as duas entradas anteriores eram 1s.

Uma entrada de 1 leva s_0 a s_1, pois agora um 1, e não dois 1s, foi lido; ela leva s_1 a s_2, pois agora dois 1s consecutivos foram lidos; e ela leva s_2 a ele mesmo, pois pelo menos dois 1s consecutivos já foram lidos. Uma entrada de 0 leva qualquer estado para s_0, pois isso quebra qualquer seqüência de 1s consecutivos. A saída para a transição de s_2 para ele próprio quando um 1 é lido, é 1, pois essa combinação de entrada e estado mostra que três 1s consecutivos já foram lidos. Todas as outras saídas são 0. O diagrama de estado para esta máquina é mostrado na Figura 6. ◀

O bit de saída final da máquina de estados finitos que construímos no Exemplo 7 é 1 se e somente se a seqüência de entrada terminar com 111. Por causa disso, dizemos que esta máquina de estados finitos **reconhece** o conjunto de seqüência de bits que termina com 111. Isso nos leva à Definição 2.

DEFINIÇÃO 2 Seja $M = (S, I, O, f, g, s_0)$ uma máquina de estados finitos e seja $L \subseteq I^*$. Dizemos que M *reconhece* (ou *aceita*) L se uma seqüência de entrada x pertence a L se e somente se o último bit de saída produzido por M quando x é dado como entrada for um 1.

TIPOS DE MÁQUINAS DE ESTADOS FINITOS Muitos tipos diferentes de máquinas de estados finitos foram desenvolvidos para modelar máquinas computacionais. Nesta seção, demos uma definição de um tipo de máquina de estados finitos. No tipo de máquina introduzido nesta seção, as saídas correspondem a transições entre estados. Máquinas desse tipo são conhecidas como **máquinas de Mealy**, pois elas foram estudadas inicialmente por G. H. Mealy, em 1955. Existe um outro tipo importante de máquina de estados finitos com saída, nas quais a saída é determinada apenas pelo estado. Esse tipo de máquina de estados finitos é conhecido como **máquina de Moore**, pois E. F. Moore o introduziu em 1956. As máquinas de Moore são consideradas em uma seqüência de exercícios no final desta seção.

No Exemplo 7, mostramos como uma máquina de Mealy pode ser usada para reconhecimento de linguagem. Entretanto, outro tipo de máquina de estados finitos, que não dá nenhuma saída, é em geral usado para esse propósito. As máquinas de estados finitos sem saída, também conhecidas como autômatos finitos, têm um conjunto de estados finais e reconhece uma seqüência se e somente se ela leva o estado inicial para o estado final. Estudaremos este tipo de máquinas de estados finitos na Seção 12.3.

Exercícios

1. Desenhe os diagramas de estados para as máquinas de estados finitos com estas tabelas de estados.

a)

	f		*g*	
	Entrada		*Entrada*	
Estado	**0**	**1**	**0**	**1**
s_0	s_1	s_0	0	1
s_1	s_0	s_2	0	1
s_2	s_1	s_1	0	0

b)

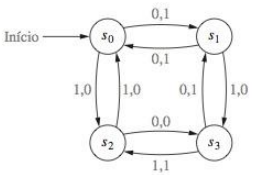
	f		*g*	
	Entrada		*Entrada*	
Estado	**0**	**1**	**0**	**1**
s_0	s_1	s_0	0	0
s_1	s_2	s_0	1	1
s_2	s_0	s_3	0	1
s_3	s_1	s_2	1	0

c)

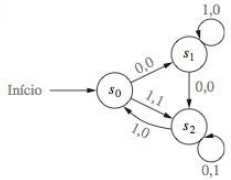
	f		*g*	
	Entrada		*Entrada*	
Estado	**0**	**1**	**0**	**1**
s_0	s_0	s_4	1	1
s_1	s_0	s_3	0	1
s_2	s_0	s_2	0	0
s_3	s_1	s_1	1	1
s_4	s_1	s_0	1	0

2. Dê as tabelas de estados para as máquinas de estados finitos com estes diagramas de estados.

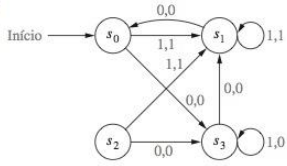
a)



b)



c)



3. Encontre a saída gerada por uma seqüência de entrada 01110 para a máquina de estados finitos com a tabela de estados dada no
- **a)** Exercício 1(a).
- **b)** Exercício 1(b).
- **c)** Exercício 1(c).

4. Encontre a saída gerada por uma seqüência de entrada 10001 para a máquina de estados finitos com o diagrama de estados dado no
- **a)** Exercício 2(a).
- **b)** Exercício 2(b).
- **c)** Exercício 2(c).

5. Encontre a saída para cada uma destas seqüências de entrada quando dadas como entrada na máquina de estados finitos do Exemplo 2.
- **a)** 0111 **b)** 11011011
- **c)** 01010101010

6. Encontre a saída para cada uma destas seqüências de entrada quando dadas como entrada na máquina de estados finitos do Exemplo 3.
- **a)** 0000 **b)** 101010
- **c)** 11011100010

7. Construa uma máquina de estados finitos que modele uma antiga máquina de refrigerante que aceita moedas de 5, 10 e 25 centavos. A máquina de refrigerante aceita troco até que 35 centavos tenham sido colocados nela. Ela devolve o troco para qualquer quantia maior do que 35 centavos. Então, o consumidor pode apertar botões para receber uma cola, um guaraná ou uma soda-limonada.

8. Construa uma máquina de estados finitos que modele uma máquina de vender jornais que tem uma porta que pode ser aberta apenas depois de três moedas de 10 centavos (e qualquer número de outras moedas) ou uma moeda de 25 centavos e uma de cinco centavos (e qualquer número de outras moedas) tiverem sido inseridas. Uma vez que a porta possa ser aberta, o consumidor abre-a e pega um jornal, fechando-a. Nenhum troco é devolvido, não importando quando dinheiro extra tenha sido inserido. O próximo consumidor começa sem nenhum crédito.

9. Construa uma máquina de estados finitos que atrase uma seqüência de entrada dois bits, dando 00 como os primeiros dois bits da saída.

10. Construa uma máquina de estados finitos que mude um a cada dois bits, começando com o segundo bit, de uma seqüência de entrada, e que deixa os outros bits sem mudança.

11. Construa uma máquina de estados finitos para o procedimento de logon em um computador, em que o usuário se loga digitando um número de identificação do usuário, que é considerado como uma entrada única, e a seguir uma senha, que é considerada como uma entrada única. Se a senha estiver incorreta, é pedido o número de identificação do usuário novamente.

12. Construa uma máquina de estados finitos para uma fechadura de combinação que contenha números de 1 a 40 e que abra apenas quando a combinação correta, 10 para a direita, segundo 8 para a esquerda e 37 para a direita, é inserida. Cada entrada é uma tripla que consiste em um número, da direção de giro e do número de vezes que a fechadura é girada naquela direção.

13. Construa uma máquina de estados finitos para uma máquina de pedágio que abre a cancela depois de 25 centavos, em moedas de 5, 10 ou 25 centavos, terem sido depositados. Não é fornecido troco para pagamentos em excesso e não é dado crédito para o próximo motorista quando mais de 25 centavos tiverem sido depositados.

14. Construa uma máquina de estados finitos para introduzir um código de segurança em um caixa eletrônico que implemente estas regras: um usuário introduz uma seqüência de quatro algarismos, um de cada vez. Se o usuário introduzir os quatro dígitos corretos da senha, o caixa eletrônico mostra uma tela de boas-vindas. Quando o usuário introduzir uma seqüência incorreta de quatro dígitos, o caixa eletrônico mostra uma tela que informa o usuário de que foi inserida uma senha incorreta. Se o usuário inserir a senha incorreta três vezes, a conta é bloqueada.

15. Construa uma máquina de estados finitos para um sistema de chaves de telefones restrito que implemente estas regras: apenas chamadas para os números de telefones 0, 911 e o algarismo 1 seguido por números de telefones com 10 algarismos que comecem com 212, 800, 866, 877 e 888 são mandados para a rede. Todas as outras seqüências de algarismos são bloqueadas pelo sistema e o usuário ouve uma mensagem de erro.

16. Construa uma máquina de estados finitos que dê uma saída 1, se o número de símbolos de entrada lido até o momento for divisível por 3, e uma saída 0, caso contrário.

17. Construa uma máquina de estados finitos que determine se a seqüência de entrada tem um 1 em sua última posição e um 0 na terceira antes da última posição lida até o momento.

18. Construa uma máquina de estados finitos que determine se a seqüência de entrada lida até o momento termina com pelo menos cinco 1s consecutivos.

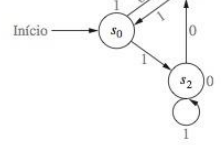
19. Construa uma máquina de estados finitos que determine se a palavra *computer* foi lida como os últimos oito caracteres da entrada lida até o momento, em que a entrada pode ser qualquer seqüência de letras do alfabeto inglês.

Uma **máquina de Moore** $M = (S, I, O, f, g, s_0)$ consiste em um conjunto finito de estados, um alfabeto de entrada I, um alfabeto de saída O, uma função de transição f que associa um próximo estado a todo par de um estado e uma entrada, uma função de saída g que associa uma saída a todo estado, e um estado inicial s_0. Uma máquina de Moore pode ser representada ou por uma tabela que liste as transições para cada par de estado e entrada e as saídas para cada estado, ou por um diagrama de estados que mostra os estados, as transições entre estados e a saída para cada estado. No diagrama, as transições são indicadas com flechas rotuladas com a entrada, e as saídas são mostradas a seguir dos estados.

20. Construa o diagrama de estados para a máquina de Moore com esta tabela de estados.

	f		
	Entrada		
Estado	**0**	**1**	***g***
s_0	s_0	s_2	0
s_1	s_3	s_0	1
s_2	s_2	s_1	1
s_3	s_2	s_0	1

21. Construa a tabela de estados para a máquina de Moore com o diagrama de estados mostrado aqui. Cada seqüência de entrada de uma máquina de Moore M produz uma seqüência de saída. Em particular, a saída correspondente à seqüência de entrada $a_1a_2 \ldots a_k$ é a seqüência $g(s_0)g(s_1) \ldots g(s_k)$, em que $s_1 = f(s_{i-1}, a_i)$ para $i = 1, 2, \ldots, k$.



22. Encontre a seqüência de saída gerada pela máquina de Moore no Exercício 20 com cada uma das seguintes seqüências de entrada.

a) 0101 **b)** 111111

c) 11101110111

23. Encontre a seqüência de saída gerada pela máquina de Moore no Exercício 21 com cada uma das seqüências de entrada do Exercício 22.

24. Construa uma máquina de Moore que dê uma saída de 1 sempre que o número de símbolos na seqüência de entrada lida até o momento seja divisível por 4.

25. Construa uma máquina de Moore que determine se uma seqüência de entrada contém um número par ou ímpar de 1s. A máquina deve dar 1 como saída se o número de 1s for par e 0 como saída se um número ímpar de 1s estiver na seqüência.

12.3 Máquinas de Estados Finitos sem Saída

Introdução

Links

Uma das mais importantes aplicações de máquinas de estados finitos é no reconhecimento de linguagem. Esta aplicação desempenha um papel fundamental no projeto e na construção de compiladores para as linguagens de programação. Na Seção 12.12, mostramos que uma máquina de estados finitos com saída pode ser usada para reconhecer uma linguagem, dando uma saída de 1 quando uma seqüência da linguagem tiver sido lida, e um 0, caso contrário. Entretanto, existem outros tipos de máquinas de estados finitos que são projetadas especificamente para o reconhecimento de linguagens. Em vez de produzir uma saída, essas máquinas têm estados finais. Uma seqüência é reconhecida se e somente se ela levar o estado inicial para um destes estados finais.

Conjunto de Seqüências

Antes de discutir máquinas de estados finitos sem saída, introduziremos algum material de revisão importante sobre conjuntos de seqüências. As operações definidas aqui serão amplamente usadas em nossa discussão sobre reconhecimento de linguagem por máquinas de estados finitos.

DEFINIÇÃO 1 Suponha que A e B sejam subconjuntos de V^*, em que V é um vocabulário. A *concatenação* de A e B, indicada por AB, é o conjunto de todas as seqüências da forma xy, em que x é uma seqüência em A e y é uma seqüência em B.

EXEMPLO 1 Seja $A = \{0, 11\}$ e $B = \{1, 10, 110\}$. Encontre AB e BA.

Solução: O conjunto AB contém toda concatenação de uma seqüência em A e uma seqüência em B. Logo, $AB = \{01, 010, 0110, 111, 11100, 11110\}$. O conjunto BA contém toda concatenação de uma seqüência em B e uma seqüência em A. Logo, $BA = \{10, 111, 100, 1011, 1100, 11011\}$. ◀

Observe que não é necessariamente verdade que $AB = BA$ quando A e B são conjuntos de V^*, em que V é um alfabeto, como ilustra o Exemplo 1.

Da definição de concatenação de dois conjuntos de seqüências, podemos definir A^n, para $n = 0, 1, 2, \ldots$. Isso é feito recursivamente, especificando que

$$A^0 = \{\lambda\},$$
$$A^{n+1} = A^n A \text{ para } n = 0, 1, 2, \ldots.$$

EXEMPLO 2 Seja $A = \{1,00\}$. Encontre A^n para $n = 0, 1, 2$ e 3.

Solução: temos $A^0 = \{\lambda\}$ e $A^1 = A^0 A = \{\lambda\} A = \{1, 00\}$. Para encontrar A^2, tomamos concatenações de pares de elementos de A. Isso fornece $A^2 = \{11, 100, 001, 0000\}$. Para encontrar A^3, tomamos concatenações de elementos em A^2 e A; isso nos dá $A^3 = \{111, 1100, 1001, 10000, 0011, 00100, 00001, 000000\}$. ◀

DEFINIÇÃO 2 Suponha que A seja um subconjunto de V^*. Então, o *fecho de Kleene* de A, indicado por A^*, é o conjunto que consiste nas concatenações de um número arbitrário de seqüências de A. Isto é, $A^* = \bigcup_{k=0}^{\infty} A^k$.

EXEMPLO 3 Quais são os fechos de Kleene dos conjuntos $A = \{0\}$, $B = \{0, 1\}$ e $C = \{11\}$?

Solução: O fecho de Kleene de A é a concatenação da seqüência 0 com ela mesma um número arbitrário finito de vezes. Portanto, $A^* = \{0^n \mid n = 0, 1, 2, \ldots\}$. O fecho de Kleene de B é a concatenação de um número arbitrário de seqüências, em que cada seqüência é um 0 ou um 1. Este é o conjunto de todas as seqüências no alfabeto $V = \{0, 1\}$. Ou seja, $B^* = V^*$. Finalmente, o fecho de Kleene de C é a concatenação da seqüência 11 com ela própria um numero arbitrário de vezes. Logo, C^* é o conjunto de todas as seqüências que consistem em um número par de 1s. Ou seja, $C^* = \{1^{2n} \mid n = 0, 1, 2, \ldots\}$. ◀

Autômatos Finitos

Daremos agora uma definição de uma máquina de estados finitos sem saída. Essas máquinas também são chamadas de **autômatos finitos**. Esta é a terminologia que usaremos para ela aqui. Essas máquinas diferem das máquinas de estados finitos estudadas na Seção 12.2 pelo fato de que elas não produzem saída, mas têm um conjunto de estados finais. Como veremos, elas reconhecem seqüências que levam o estado inicial para um estado final.

DEFINIÇÃO 3 Um *autômato finito* $M = (S, I, f, s_0, F)$ consiste em um conjunto finito S de *estados*, um *alfabeto de entrada* finito I, uma *função de transição* f que associa um próximo estado a cada par de estado e entrada (de modo que $f: S \times I \to S$), um *estado inicial* s_0 e um subconjunto F de S que consiste nos *estados finais* (ou estados *aceitos*).

Podemos representar autômatos finitos usando tabelas de estados ou diagramas de estados. Os estados finais são indicados nos diagramas de estados por meio de círculos duplos.

EXEMPLO 4 Construa o diagrama de estados para o autômato finito $M = (S, I, f, s_0, F)$, em que $S = \{s_0, s_1, s_2, s_3\}$, $I = \{0, 1\}$, $F = \{s_0, s_3\}$ e a função de transição f é dada na Tabela 1.

Solução: O diagrama de estado é mostrado na Figura 1. Observe que como ambas as entradas 0 e 1 levam s_2 a s_0, escrevemos 0, 1 sobre a aresta de s_2 a s_0. ◀

ESTENDENDO A FUNÇÃO DE TRANSIÇÃO A função de transição f de uma máquina de estados finitos $M = (S, I, f, s_0, F)$ pode ser estendida de modo que ela seja definida para todos os pares de estados e seqüências; ou seja, f pode ser estendida para uma função $f: S \times I^* \to S$. Seja $x = x_1x_2 \ldots x_k$ uma seqüência em I^*. Então $f(s_1, x)$ é o estado obtido pelo uso de cada símbolo sucessivo de x da esquerda para direita, como entrada, começando com o estado s_1. De

Links

STEPHEN COLE KLEENE (1909–1994) Stephen Kleene nasceu em Hartford, Connecticut. Sua mãe, Alice Lena Cole, era poetisa, e seu pai, Gustav Adolph Kleene, era professor de economia. Kleene freqüentou o Amherst College e obteve seu Ph.D. em Princeton, em 1934, onde estudou sob a orientação do famoso lógico Alonzo Church. Kleene entrou para o corpo docente da universidade de Wisconsin em 1935, onde permaneceu exceto por diversos afastamentos, incluindo diversas estadas no Instituto de Estudos Avançados, em Princeton. Durante a segunda guerra mundial, ele foi instrutor de navegação na Naval Reserve's Midshipman's School e mais tarde foi diretor do Naval Research Laboratory. Kleene fez contribuições significantes para a teoria das funções recursivas, investigando questões de computabilidade e decidibilidade, e demonstrou um dos resultados centrais da teoria de autômatos. Ele foi Diretor Interino do Mathematics Research Center e reitor do College of Letters and Sciences na universidade de Wisconsin. Kleene era um estudioso da história natural. Ele descobriu uma variedade de borboleta que ainda não havia sido descrita e que leva seu nome. Ele era um excursionista e um alpinista entusiasmado. Kleene também se distinguiu como um talentoso contador de anedotas, usando uma voz poderosa que podia ser ouvida diversas salas adiante.

TABELA 1		
	f	
	Entrada	
Estado	**0**	**1**
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_0
s_3	s_2	s_1

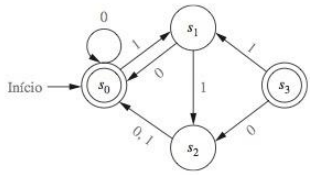


FIGURA 1 O Diagrama de Estado para um Autômato Finito.

s_1 prosseguimos para o estado $s_2 = f(s_1, x_1)$, a seguir para o estado $s_3 = f(s_2, x_2)$, e assim por diante, com $f(s_1, x) = f(s_k, x_k)$. Formalmente, podemos definir esta função de transição estendida f recursivamente para a máquina de estados finitos determinística $M = (S, I, f, s_0, F)$ por

(*i*) $f(s, \lambda) = s$ para todo estado $s \in S$; e
(*ii*) $f(s, xa) = f(f(s, x), a)$ para todo $s \in S$, $x \in I^*$, e $a \in I$.

Podemos usar indução estrutural e essa definição recursiva para demonstrar propriedades dessa função de transição estendida. Por exemplo, no Exercício 15, pedimos que você demonstre que

$$f(s, xy) = f(f(s, x), y)$$

para todo estado $s \in S$ e seqüências $x \in I^*$ e $y \in I^*$.

Reconhecimento de Linguagem por Máquinas de Estados Finitos

A seguir, definimos alguns termos que são usados no estudo de reconhecimento por autômatos finitos de determinados conjuntos de seqüências.

DEFINIÇÃO 4 Dizemos que uma seqüência x pode ser *reconhecida* ou *aceita* por uma máquina $M = (S, I, f, s_0, F)$ se ela leva o estado inicial s_0 a um estado final, ou seja, $f(s_0, x)$ é um estado em F. A *linguagem reconhecida* ou *aceita* pela máquina M, indicada por $L(M)$, é o conjunto de todas as seqüências que são reconhecidas por M. Dois autômatos finitos são denominados *equivalentes* se eles reconhecem a mesma linguagem.

No Exemplo 5, encontraremos uma linguagem reconhecida por diversos autômatos finitos.

EXEMPLO 5 Determine as linguagens reconhecidas pelos autômatos finitos M_1, M_2, e M_3 na Figura 2.

Solução: O único estado final de M_1 é s_0. As seqüências que levam s_0 a ele próprio são aquelas que consistem em zero ou mais 1s consecutivos. Logo, $L(M_1) = \{1^n \mid n = 0, 1, 2, \ldots\}$.

O único estado final de M_2 é s_2. As únicas seqüências que levam s_0 a s_2 são 1 e 01. Logo, $L(M_2) = \{1, 01\}$.

Os estados finais de M_3 são s_0 e s_3. As únicas seqüências que levam s_0 a ele próprio são λ, 0, 00, 000, . . . , ou seja, qualquer seqüência de zero ou mais 0s consecutivos. As únicas seqüências que levam s_0 a s_3 são uma seqüência de zero ou mais 0s consecutivos seguidos por 10, seguido

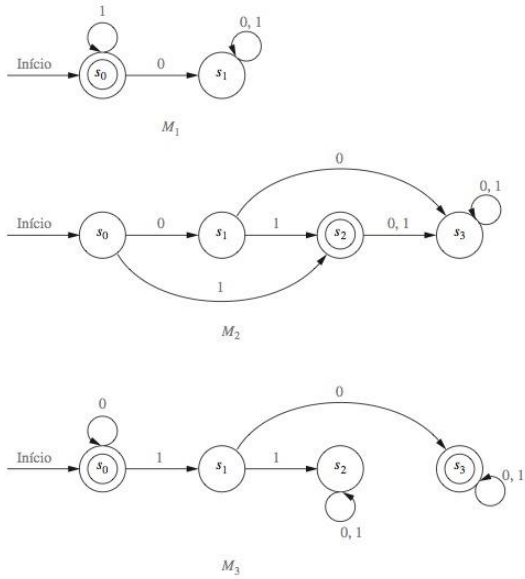


FIGURA 2 Alguns Autômatos Finitos

por qualquer seqüência. Logo, $L(M_3) = \{0^n, 0^n10x \mid n = 0, 1, 2, \ldots,$ e x é qualquer seqüência$\}$. ◀

PROJETANDO AUTÔMATOS FINITOS Podemos, em geral, construir um autômato finito que reconheça um dado conjunto de seqüências, adicionando cuidadosamente estados e transições e determinando quais desses estados deveriam ser estados finais. Quando for apropriado, incluiremos estados que podem guardar algumas das propriedades da seqüência de entrada, fornecendo uma memória limitada ao autômato finito. Os exemplos 6 e 7 ilustram algumas das técnicas que podem ser usadas para construir autômatos finitos que reconhecem tipos particulares de conjuntos de seqüências.

EXEMPLO 6 Construa autômatos finitos determinísticos que reconheçam cada uma destas linguagens.

Exemplos Extras

(a) O conjunto das seqüências de bits que começam com dois 0s.
(b) O conjunto das seqüências de bits que contêm dois 0s consecutivos.
(c) O conjunto das seqüências de bits que não contêm dois 0s consecutivos.
(d) O conjunto das seqüências de bits que terminam com dois 0s.
(e) O conjunto das seqüências de bits que contêm pelo menos dois 0s.

Solução: (a) Nosso objetivo é construir um autômato finito determinístico que reconheça o conjunto de seqüências de bits que começam com dois 0s. Além do estado inicial s_0, incluímos um estado não final s_1; movemos de s_0 para s_1 se o primeiro bit for um 0, a seguir adicionamos o estado final s_2, para o qual nos movemos a partir de s_1 se o segundo bit for um 0. Quando atingirmos s_1, saberemos que os dois primeiros bits de entrada são ambos 0s, de modo que permanecemos no estado s_2 não importa o que aconteça nos bits seguintes (se houver algum). Então, nos

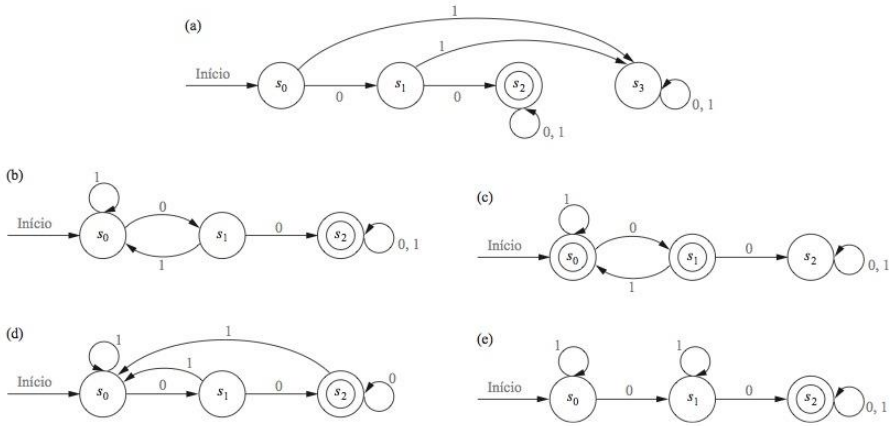


FIGURA 3 Autômatos Finitos Determinísticos que Reconhecem as Linguagens no Exemplo 6.

movemos para um estado não final s_3 a partir de s_0, se o primeiro bit for um 1 e, a partir de s_1, se o segundo bit for um 1. O leitor deve verificar que o autômato finito na Figura 3(a) reconhece o conjunto de seqüências de bits que começam com dois 0s.

(b) Nosso objetivo é construir um autômato finito determinístico que reconheça o conjunto de seqüências de bits que contêm dois 0s consecutivos. Além do estado inicial s_0, incluímos um estado não final s_1, que nos diz que o último bit de entrada visto é um 0, mas ou o bit antes dele era um 1 ou este bit era o bit inicial da seqüência. Incluímos um estado final s_2 para o qual nos movemos a partir de s_1 quando o próximo bit de entrada depois de um 0 também for um 0. Se um 1 seguir um 0 na seqüência (antes de encontrarmos dois 0s consecutivos), voltamos a s_0 e começamos a procurar por 0s consecutivos novamente. O leitor deve verificar que o autômato finito na Figura 3(b) reconhece o conjunto de seqüências de bits que contêm dois 0s consecutivos.

(c) Nosso objetivo é construir um autômato finito determinístico que reconheça o conjunto de seqüências de bits que não contêm dois 0s consecutivos. Além do estado inicial s_0, que deve ser um estado final, incluímos um estado final s_1, para o qual nos movemos a partir de s_0 quando 0 for o primeiro bit de entrada. Quando um bit de entrada for um 1, voltamos para o estado s_0 ou ficamos nele. Adicionamos um estado s_2 para o qual nos movemos a partir de s_1 quando o bit de entrada for um 0. Atingir s_2 nos diz que vimos dois 0s consecutivos como bits de entrada. Permanecemos no estado s_2 uma vez que o tenhamos atingido; este estado não é final. O leitor deve verificar que o autômato finito na Figura 3(c) reconhece o conjunto de seqüências de bits que não contêm dois 0s consecutivos. [O leitor perspicaz observará a relação entre o autômato finito construído aqui e o construído na parte (b). Veja o Exercício 39.]

(d) Nosso objetivo é construir um autômato finito determinístico que reconheça o conjunto de seqüências de bits que terminam com dois 0s. Além do estado inicial s_0, incluímos um estado não final s_1, para o qual nos movemos se o primeiro bit for 0. Incluímos um estado final s_2, para o qual nos movemos a partir de s_1 se o próximo bit de entrada depois de um 0 também for um 0. Se uma entrada 0 seguir um 0 anterior, permanecemos no estado s_2, pois os dois últimos bits de entrada ainda são 0s. Uma vez que estejamos no estado s_2, um bit de entrada de 1 nos manda de volta para s_0 e começamos a procurar por 0s consecutivos de novo. Também voltamos para s_0 se a próxima entrada for um 1 quando estivermos no estado s_1. O leitor deve verificar que o autô-

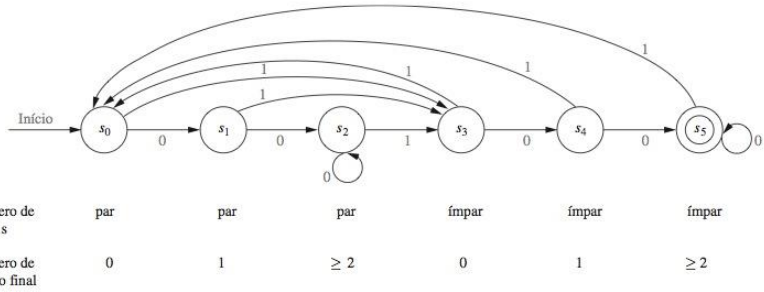


FIGURA 4 Autômato Finito Determinístico que Reconhece o Conjunto de Seqüências de Bits que Contêm Um Número Ímpar de 1s e Termina com pelo menos Dois 0s.

mato finito na Figura 3(d) reconhece o conjunto de seqüências de bits que terminam com dois 0s.

(e) Nosso objetivo é construir um autômato finito determinístico que reconhece o conjunto de seqüências de bits que contenham dois 0s. Além do estado inicial, incluímos um estado s_1 que não é final; permanecemos em s_0 até que um bit de entrada seja um 0 e nos movemos para s_1 quando encontramos o primeiro bit 0 na entrada. Adicionamos um estado final s_2 para o qual nos movemos a partir de s_1 uma vez que encontremos um segundo bit 0. Sempre que encontrarmos um 1 como entrada, permanecemos no estado corrente. Uma vez que tenhamos atingido s_2, permanecemos lá. Aqui, s_1 e s_2 são usados para nos dizer que já vimos um ou dois 0s na seqüência de entrada até agora, respectivamente. O leitor deve verificar que o autômato finito na Figura 3(e) reconhece o conjunto de seqüências de bits que contêm dois 0s. ◀

EXEMPLO 7 Construa um autômato finito determinístico que reconheça o conjunto de seqüências de bits que contêm um número ímpar de 1s e que termina com pelo menos dois 0s consecutivos.

Solução: Podemos construir um autômato finito determinístico que reconheça conjuntos especificados por meio da inclusão de estados que guardem tanto a paridade do número de bits 1 quanto de se encontrar zero, um ou pelo menos dois 0s no final da seqüência de entrada.

O estado inicial s_0 pode ser usado para nos dizer que a entrada lida até agora contém um número par de 1s e termina sem 0s (isto é, está vazia ou termina com um 1). Além do estado inicial, incluímos mais cinco estados. Então, nos movemos para os estados s_1, s_2, s_3, s_4 e s_5, respectivamente, quando a seqüência de entrada lida até o momento contém um número par de 1s e termina com um 0; quando ela contém um número par de 1s e termina com pelo menos dois 0s; quando ela contém um número ímpar de 1s e termina sem nenhum 0; quando ela contém um número ímpar de 1s e termina com um 0; e quando ela contém um número ímpar de 1s e termina com dois 0s. O estado s_5 é um estado final.

O leitor deve verificar que o autômato finito na Figura 4 reconhece o conjunto de seqüências de bits que contêm um número ímpar de 1s e termina com pelo menos dois 0s consecutivos. ◀

AUTÔMATOS FINITOS EQUIVALENTES Na Definição 4 especificamos que dois autômatos finitos são equivalentes se eles reconhecem a mesma linguagem. O Exemplo 8 fornece um exemplo de duas máquinas de estados finitos determinísticas equivalentes.

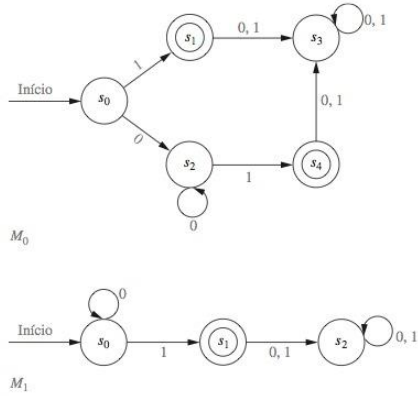


FIGURA 5 M_0 **e** M_1 **São Autômatos Finitos Equivalentes.**

EXEMPLO 8 Mostre que os dois autômatos finitos M_0 e M_1 apresentados na Figura 5 são equivalentes.

Solução: Para que uma seqüência x seja reconhecida por M_0, x deve nos levar de s_0 para o estado final s_1 ou para o estado final s_4. A única seqüência que nos leva de s_0 para s_1 é a seqüência 1. As seqüências que nos levam de s_0 para s_4 são as seqüências que começam com um 0, o que nos leva de s_0 para s_2, seguido por um ou mais 0s adicionais, o que mantém a máquina no estado s_2, seguido por um 1, o que nos leva do estado s_2 para o estado final s_4. Todas as outras seqüências nos levam de s_0 para um estado que não é final. (Deixamos os detalhes para o leitor.) Concluímos que $L(M_0)$ é o conjunto de seqüências de zero ou mais bits 0 seguidos por um 1 final.

Para que uma seqüência x seja reconhecida por M_1, x deve nos levar de s_0 para o estado final s_1. Assim, para que x seja reconhecida, ela deve começar com algum número de 0s, o que nos deixa no estado s_0, seguido por um 1, o que nos leva para o estado final s_1. Uma seqüência só de 0s não é reconhecida, pois nos deixa no estado s_0, que não é final. Todas as seqüências que contêm um 0 depois de um 1 não são reconhecidas, pois nos levam para o estado s_2, que não é final. Segue que $L(M_1)$ é o mesmo que $L(M_0)$. Concluímos que M_0 e M_1 são equivalentes.

Observe que a máquina de estados finitos M_1 tem apenas três estados. Nenhuma máquina de estados finitos com menos de três estados pode ser usada para reconhecer o conjunto de todas as seqüências de zero ou mais bits 0 seguida de um 1 (veja o Exercício 37). ◀

Como mostra o Exemplo 8, um autômato finito pode ter mais estados do que um equivalente a ele. De fato, os algoritmos usados para construir autômatos finitos para reconhecer determinadas linguagens podem ter mais estados do que o necessário. O uso de máquinas de estados finitos desnecessariamente grandes para reconhecer linguagens pode tornar as aplicações tanto em hardware quanto em software ineficientes e caras. Esse problema aparece quando autômatos finitos são usados em compiladores, que traduzem programas computacionais para uma linguagem que o computador possa entender (código objeto).

Os exercícios 58 a 61 desenvolvem um procedimento que constrói um autômato finito com o menor número possível de estados entre todos os autômatos finitos equivalentes a um dado autômato finito. Este procedimento é conhecido como **minimização de máquina**. O procedimento de minimização descrito nesses exercícios reduz o número de estados, substituindo-os por classes de equivalência de estados com respeito a uma relação de equivalência na qual dois estados são equivalentes se toda seqüência de entrada manda ambos os estados para um estado

final ou manda ambos para um estado que não é final. Antes de o procedimento de minimização começar, todos os estados que não podem ser atingidos a partir do estado inicial usando qualquer seqüência de entrada são removidos primeiro; removê-los não muda a linguagem reconhecida.

Autômatos Finitos Não Determinísticos

Os autômatos finitos discutidos até agora são **determinísticos**, pois para cada par de estado e valor de entrada existe um único estado seguinte dado pela função de transição. Existe outro tipo importante de autômato finito no qual pode haver diversos estados seguintes possíveis para cada par de valor de entrada e estado. Essas máquinas são chamadas de **não determinísticas**. Autômatos finitos não determinísticos são importantes na determinação de quais linguagens podem ser reconhecidas por um autômato finito.

DEFINIÇÃO 5

Links

Um *autômato finito não determinístico* $M = (S, I, f, s_0, F)$ consiste em um conjunto S de estados, um alfabeto de entrada I, uma função de transição f que associa um conjunto de estados a cada par de estado e entrada (de modo que $f : S \times I \rightarrow P(S)$), um estado inicial s_0 e um subconjunto F de S que consiste nos estados finais.

Podemos representar um autômato finito não determinístico usando tabelas de estados ou diagramas de estados. Quando usamos uma tabela de estados, para cada par de estado e valor de entrada damos uma lista de possíveis estados seguintes. No diagrama de estados, incluímos uma

GRACE BREWSTER MURRAY HOPPER (1906–1992) Grace Hopper nasceu na cidade de Nova York; quando criança mostrou uma intensa curiosidade sobre como as coisas funcionam. Com sete anos, desmontava despertadores para descobrir seus mecanismos. Ela herdou seu amor pela matemática de sua mãe, que recebeu permissão especial para estudar geometria (mas não álgebra e trigonometria) em uma época na qual as mulheres eram ativamente desencorajadas em relação a esses estudos. Hopper se inspirou em seu pai, um agente de seguros de sucesso, que perdeu suas pernas por problemas circulatórios. Ele disse a seus filhos que poderiam fazer qualquer coisa que realmente quisessem. Ele inspirou Hopper a buscar uma educação de alto nível e a não se conformar com o papel feminino usual. Seus pais garantiram que ela tivesse uma excelente educação, freqüentando escolas particulares para mulheres em Nova York. Hopper ingressou no Vassar College em 1924, onde estudou matemática e física, tendo se formado em 1928. Ela obteve o título de mestre em matemática da Universidade de Yale em 1930. Nesse mesmo ano, ela se casou com um professor da New York School of Commerce; mais tarde ela se divorciou e não teve filhos. Hopper foi professora de matemática em Vassar de 1931 a 1943, tendo obtido o título Ph.D. em Yale, em 1934.

Depois do ataque a Pearl Harbor, Hopper, que vinha de uma família com forte tradição militar, decidiu deixar sua posição acadêmica e ingressar na Navy WAVES. Para se alistar, ela precisou de permissão especial para deixar sua posição estratégica de professora de matemática, bem como de uma concessão por pesar muito pouco. Em dezembro de 1943, ela ingressou na Navy Reserve e recebeu treinamento na Midshipman´s School for Women. Hopper foi designada para trabalhar na Naval Ordnance Laboratory na Universidade de Harvard. Ela escreveu programas para o primeiro computador digital seqüencial em grande escala do mundo, que foi usado para ajudar a apontar a artilharia naval em climas variáveis. Credita-se a Hopper a invenção do termo "bug" para se referir a falhas de hardware, mas ele era usado em Harvard antes de sua chegada lá. Entretanto, é verdade que Hopper e seu time de programação encontraram uma traça em um dos condutores no hardware do computador que fez o sistema cair. Esta famosa traça foi colada em um livro do laboratório. Na década de 1950, Hooper inventou o termo "debug" para o processo de remover erros de programação.

Em 1946, quando a marinha disse a ela que era muito velha para o serviço ativo, Hopper decidiu permanecer em Harvard como uma assistente de pesquisa civil. Em 1949, ela deixou Harvard para ingressar na Eckert-Mauchly Computer Corporation, onde ajudou a desenvolver o primeiro computador comercial, o UNIVAC. Hopper permaneceu com esta companhia quando ela foi incorporada pela Remington Rand e quando a Remington Rand se fundiu com a Sperry Corporation. Hopper era uma visionária do poder em potencial dos computadores; ela entendeu que estes se tornariam amplamente usados se pudessem ser desenvolvidas ferramentas que fossem amigáveis tanto ao usuário quanto às aplicações. Em particular, ela acreditava que os programas computacionais poderiam ser escritos em inglês, em vez de usar instruções de máquina. Para ajudar a atingir esse objetivo, ela desenvolveu o primeiro compilador e publicou o primeiro artigo de pesquisa em compiladores em 1952. Hopper também é conhecida como a mãe da linguagem computacional COBOL; membros da sua equipe ajudaram a desenvolver o projeto básico da linguagem COBOL usando seu trabalho anterior como base.

Em 1966, Hopper se aposentou da Navy Reserve. Entretanto, apenas sete meses mais tarde, a marinha a chamou da aposentadoria para ajudar a padronizar as linguagens de computação de alto nível da marinha. Em 1983, ela foi promovida à patente de Comodoro por indicação especial do presidente e, em 1985, foi promovida à patente de Rear Admiral. Sua aposentadoria da marinha, aos 80 anos de idade, se deu a bordo do *USS Constitution*.

TABELA 2		
	f	
	Entrada	
Estado	**0**	**1**
s_0	s_0, s_1	s_3
s_1	s_0	s_1, s_3
s_2		s_0, s_2
s_3	s_0, s_1, s_2	s_1

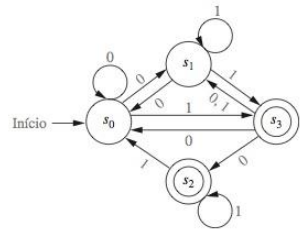


FIGURA 6 O Autômato Finito Não Determinístico com Tabela de Estados Mostrada na Tabela 2.

aresta de cada estado para todos os estados seguintes possíveis, rotulando as arestas com a entrada ou as entradas que levam a esta transição.

EXEMPLO 9 Encontre o diagrama de estados para o autômato finito não determinístico com tabela de estados mostrada na Tabela 2. Os estados finais são s_2 e s_3.

Solução: O diagrama de estados para este autômato é mostrado na Figura 6. ◀

EXEMPLO 10 Encontre a tabela de estados para o autômato finito não determinístico com o diagrama de estado mostrado na Figura 7.

Solução: A tabela de estados é dada na Tabela 3. ◀

O que significa um autômato finito não determinístico reconhecer uma seqüência $x = x_1x_2 \ldots x_k$? O primeiro símbolo de entrada x_1 leva o estado inicial s_0 a um conjunto S_1 de estados. O próximo símbolo de entrada x_2 leva cada um dos estados em S_1 a um conjunto de estados. Seja S_2 a união destes conjuntos. Continuamos o processo, incluindo em cada estágio todos os estados obtidos usando um estado obtido no estágio anterior e o símbolo de entrada corrente. **Reconhecemos**, ou **aceitamos**, a seqüência x se existir um estado final no conjunto de todos os estados que podem ser obtidos a partir de s_0 usando x. A **linguagem reconhecida** por um autômato finito não determinístico é o conjunto de todas as seqüências reconhecidas por este autômato.

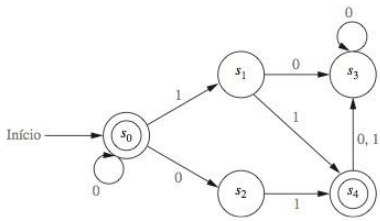


FIGURA 7 Autômato Finito Não Determinístico.

TABELA 3		
	f	
	Entrada	
Estado	**0**	**1**
s_0	s_0, s_2	s_1
s_1	s_3	s_4
s_2		s_4
s_3	s_3	
s_4	s_3	s_3

EXEMPLO 11 Encontre a linguagem reconhecida pelo autômato finito não determinístico mostrado na Figura 7.

Solução: Como s_0 é um estado final, e existe uma transição de s_0 para ele mesmo quando a entrada é 0, a máquina reconhece todas as seqüências que consistem em zero ou mais 0s consecutivos. Além disso, como s_4 é um estado final, qualquer seqüência que tenha s_4 no conjunto de estados que podem ser atingidos a partir de s_0 com esta seqüência de entrada pode ser reconhecida. As únicas destas seqüências são as seqüências que consistem em zero ou mais 0s consecutivos seguidos por 01 ou 11. Como s_0 e s_4 são os únicos estados finais, a linguagem reconhecida pela máquina é $\{0^n, 0^n01, 0^n11 \mid n \geq 0\}$. ◀

Um fato importante é que uma linguagem reconhecida por um autômato finito não determinístico também é reconhecida por um autômato finito determinístico. Tiraremos vantagem deste fato na Seção 12.4, quando determinaremos quais linguagens são reconhecidas por autômatos finitos.

TEOREMA 1 Se a linguagem L é reconhecida por um autômato finito não determinístico M_0, então L também é reconhecida por um autômato finito determinístico M_1.

Demonstração: Descreveremos como construir o autômato finito determinístico M_1 que reconhece L a partir de M_0, o autômato finito não determinístico que reconhece esta linguagem. Cada estado em M_1 será composto de um conjunto de estados em M_0. O símbolo inicial de M_1 é $\{s_0\}$, que é o conjunto que contém o estado inicial em M_0. O conjunto de entrada de M_1 é o mesmo que o conjunto de entrada de M_0.

Dado um estado $\{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\}$ de M_1, o símbolo de entrada x leva este estado à união dos conjuntos dos estados seguintes para os elementos neste conjunto, ou seja, a união dos conjuntos $f(s_{i_1}), f(s_{i_2}), \ldots, f(s_{i_k})$. Os estados de M_1 são todos os subconjuntos de S, o conjunto de estados de M_0, que são obtidos desta forma, começando com s_0. (Existem até 2^n estados na máquina determinística, em que n é o número de estados na máquina não determinística, pois todos os subconjuntos podem ocorrer como estados, incluindo o conjunto vazio, embora em geral ocorram muito menos estados.) Os estados finais de M_1 são aqueles conjuntos que contêm um estado final de M_0.

Suponha que uma seqüência de entrada seja reconhecida por M_0. Então, um dos estados que podem ser atingidos a partir de s_0 usando esta seqüência de entrada é um estado final (o leitor deve fornecer uma demonstração indutiva disso). Isso significa que em M_1, esta seqüência de entrada leva de $\{s_0\}$ para um conjunto de estados de M_0 que contém um estado final. Este subconjunto é um estado final de M_1, de modo que esta seqüência também é reconhecida por M_1. Além disso, uma seqüência de entrada que não seja reconhecida por M_0 não leva a nenhum estado final de M_0. (O leitor deve fornecer os detalhes da demonstração desta afirmação.) Conseqüentemente, esta seqüência de entrada não leva de $\{s_0\}$ a um estado final de M_1. ◁

EXEMPLO 12 Encontre um autômato finito determinístico que reconheça a mesma linguagem que o autômato finito não determinístico do Exemplo 10.

Solução: O autômato determinístico mostrado na Figura 8 é construído a partir do autômato não determinístico do Exemplo 10. Os estados deste autômato determinístico são subconjuntos do conjunto de todos os estados da máquina não determinística. O próximo estado de um subconjunto sob um símbolo de entrada é o subconjunto com os próximos estados na máquina não determinística de todos os elementos neste subconjunto. Por exemplo, com uma entrada 0, $\{s_0\}$ vai para $\{s_0, s_2\}$, pois s_0 tem transições para si mesmo e para s_2 na máquina não determinística; o conjunto $\{s_0, s_2\}$ vai para $\{s_1, s_4\}$ com a entrada 1, pois s_0 vai apenas para s_1 e s_2 vai apenas para s_4 com a entrada 1 na máquina não determinística; e o conjunto $\{s_1, s_4\}$ vai para $\{s_3\}$ com a entrada 0, pois s_1 e s_4 vão ambos apenas para s_3 com a entrada 0 na máquina determinística. Todos os subconjuntos que são obtidos desse modo são incluídos na máquina de estados finitos determinística. Observe que o conjunto vazio é um dos estados desta máquina, pois ele é o subconjunto que contém todos os estados seguintes a $\{s_3\}$ com a entrada 1. O estado inicial é $\{s_0\}$ e o conjunto de estados finais são todos aqueles que incluem s_0 ou s_4. ◀

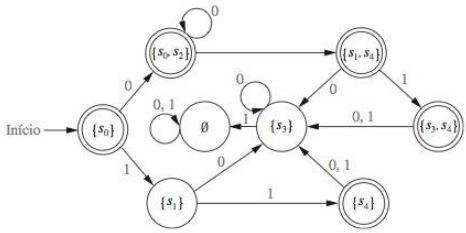


FIGURA 8 Autômato Determinístico Equivalente ao Autômato Não Determinístico do Exemplo 10.

Exercícios

1. Sejam $A = \{0, 11\}$ e $B = \{00, 01\}$. Encontre cada um destes conjuntos.
a) AB **b)** BA **c)** A^2 **d)** B^3

2. Mostre que se A é um conjunto de seqüências, então $A\emptyset = \emptyset A = \emptyset$.

3. Encontre todos os pares de conjuntos de seqüências A e B para os quais $AB = \{10, 111, 1010, 1000, 10111, 101000\}$.

4. Mostre que estas igualdades são válidas.
a) $\{\lambda\}^* = \{\lambda\}$
b) $(A^*)^* = A^*$ para todo conjunto A de seqüências.

5. Descreva os elementos do conjunto A^* para estes valores de A.
a) $\{10\}$ **b)** $\{111\}$ **c)** $\{0, 01\}$ **d)** $\{1, 101\}$

6. Seja V um alfabeto e sejam A e B subconjuntos de V^*. Mostre que $|AB| \leq |A||B|$.

7. Seja V um alfabeto e sejam A e B subconjuntos de V^*, com $A \subseteq B$. Mostre que $A^* \subseteq B^*$.

8. Suponha que A seja um subconjunto de V^*, em que V é um alfabeto. Demonstre ou mostre que não é válida cada uma destas afirmações.
a) $A \subseteq A^2$ **b)** se $A = A^2$, então $\lambda \in A$
c) $A\{\lambda\} = A$ **d)** $(A^*)^* = A^*$
e) $A^*A = A^*$ **f)** $|A^n| = |A|^n$

9. Determine se a seqüência 11101 está em cada um destes conjuntos.
a) $\{0, 1\}^*$ **b)** $\{1\}^*\{0\}^*\{1\}^*$
c) $\{11\}\ \{0\}^*\{01\}$ **d)** $\{11\}^*\{01\}^*$
e) $\{111\}^*\{0\}^*\{1\}$ **f)** $\{11, 0\}\ \{00,101\}$

10. Determine se a seqüência 01001 está em cada um destes conjuntos.
a) $\{0, 1\}^*$ **b)** $\{0\}^*\{10\}\{1\}^*$
c) $\{010\}^*\{0\}^*\{1\}$ **d)** $\{010, 011\}\ \{00, 01\}$
e) $\{00\}\{0\}^*\{01\}$ **f)** $\{01\}^*\{01\}^*$

11. Determine se cada uma destas seqüências é reconhecida pelo autômato finito determinístico da Figura 1.
a) 111 **b)** 0011 **c)** 1010111 **d)** 011011011

12. Determine se cada uma destas seqüências é reconhecida pelo autômato finito determinístico da Figura 1.
a) 010 **b)** 1101 **c)** 1111110 **d)** 010101010

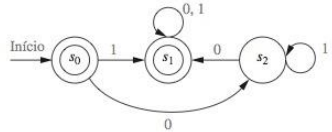
13. Determine se todas as seqüências em cada um destes conjuntos são reconhecidas pelo autômato finito determinístico na Figura 1.
a) $\{0\}^*$ **b)** $\{0\}\ \{0\}^*$ **c)** $\{1\}\ \{0\}^*$
d) $\{01\}^*$ **e)** $\{0\}^*\{1\}^*$ **f)** $\{1\}\{0, 1\}^*$

14. Mostre que se $M = (S, I, f, s_0, F)$ for um autômato finito determinístico e $f(s, x) = s$ para um estado $s \in S$ e uma seqüência de entrada $x \in I^*$, então $f(s, x^n) = s$ para todo inteiro não negativo n. (Aqui, x^n é a concatenação de n cópias da seqüência x, definida recursivamente no Exercício 37 da Seção 4.3.)

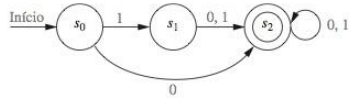
15. Dado um autômato finito determinístico $M = (S, I, f, s_0, F)$, use indução estrutural e definição recursiva de função de transição estendida f para demonstrar que $f(s, xy) = f(f(s, x), y)$ para todos os estados $s \in S$ e todas as seqüências $x \in I^*$ e $y \in I^*$.

Nos exercícios 16 a 22, encontre a linguagem reconhecida pelo autômato finito determinístico dado.

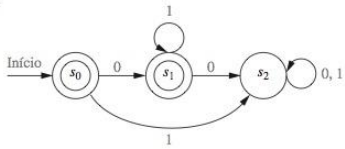
16.



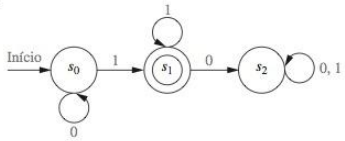
17.



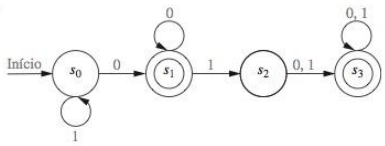
18.



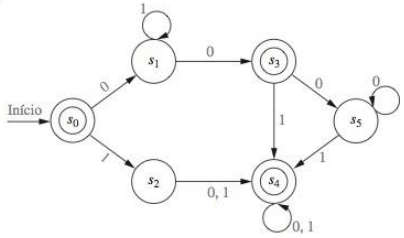
19.



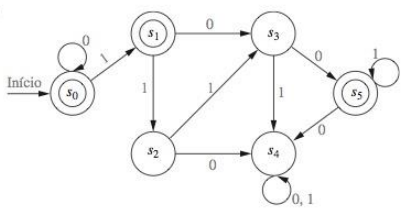
20.



21.



22.



23. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que começam com 01.

24. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que terminam com 10.

25. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que contenham a seqüência 101.

26. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que não contenham três 0s consecutivos.

27. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que contenham exatamente três 0s.

28. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que contenham pelo menos três 0s.

29. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que contenham três 1s consecutivos.

30. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que comecem com 0 ou com 11.

31. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que comecem e terminem com 11.

32. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que contenham um número par de 1s.

33. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que contenham um número ímpar de 0s.

34. Construa um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que contenham um número par de 0s e um número ímpar de 1s.

35. Construa um autômato finito que reconheça o conjunto das seqüências de bits que consistem em um 0 seguido por uma seqüência com um número ímpar de 1s.

36. Construa um autômato finito com quatro estados que reconheça o conjunto das seqüências de bits que contenham um número par de 1s e um número ímpar de 0s.

37. Mostre que não existe um autômato finito com dois estados que reconheça o conjunto de todas as seqüências de bits que têm um ou mais bits 1s e termina com um 0.

38. Mostre que não existe um autômato finito com três estados que reconheça o conjunto das seqüências de bits que contêm um número par de 1s e um número par de 0s.

39. Explique como você pode mudar um autômato finito determinístico M de modo que o autômato mudado reconheça o conjunto $I^* - L(M)$.

40. Use o Exercício 39 e o autômato finito construído no Exemplo 6 para encontrar o autômato finito determinístico que reconheça cada um destes conjuntos.

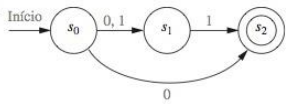
- **a)** o conjunto das seqüências de bits que não começam com dois 0s
- **b)** o conjunto das seqüências de bits que não terminam com dois 0s
- **c)** o conjunto das seqüências de bits que contêm no máximo um 0 (ou seja, que não contêm pelo menos dois 0s)

41. Use o procedimento que você descreveu no Exercício 39 e o autômato finito que você construiu no Exercício 25 para encontrar um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que não contêm a seqüência 101.

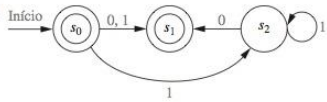
42. Use o procedimento que você descreveu no Exercício 39 e o autômato finito que você construiu no Exercício 29 para encontrar um autômato finito determinístico que reconheça o conjunto de todas as seqüências de bits que não contêm três 1s consecutivos.

Nos exercícios 43 a 49, encontre a linguagem reconhecida pelo autômato finito não determinístico dado.

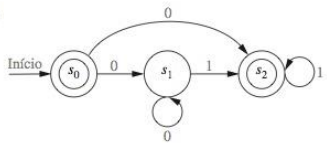
43.



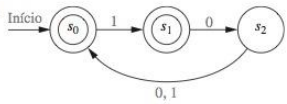
44.



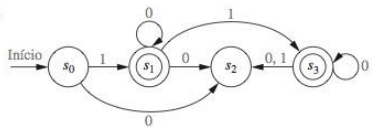
45.



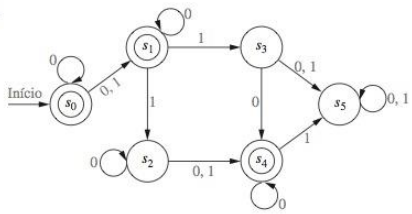
46.



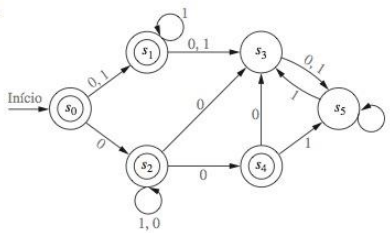
47.



48.



49.



50. Encontre um autômato finito determinístico que reconheça a mesma linguagem que o autômato finito não determinístico do Exercício 43.

51. Encontre um autômato finito determinístico que reconheça a mesma linguagem que o autômato finito não determinístico do Exercício 44.

52. Encontre um autômato finito determinístico que reconheça a mesma linguagem que o autômato finito não determinístico do Exercício 45.

53. Encontre um autômato finito determinístico que reconheça a mesma linguagem que o autômato finito não determinístico do Exercício 46.

54. Encontre um autômato finito determinístico que reconheça a mesma linguagem que o autômato finito não determinístico do Exercício 47.

55. Encontre um autômato finito determinístico que reconheça cada um destes conjuntos.

a) $\{0\}$ **b)** $\{1, 00\}$ **c)** $\{1^n \mid n = 2, 3, 4, \ldots\}$

56. Encontre um autômato finito não determinístico que reconheça cada uma das linguagens do Exercício 27 e tenha menos estados, se possível, que o autômato determinístico que você encontrou naquele exercício.

***57.** Mostre que não existe nenhum autômato finito que reconheça o conjunto de seqüências de bits que contêm o mesmo número de 0s e de 1s.

Nos exercícios 58 a 62, introduzimos uma técnica para a construção de uma máquina de estados finitos determinística equivalente a determinada máquina de estados finitos determinística com o menor número possível de estados. Suponha que $M = (S, I, f, s_0, F)$ seja um autômato finito e que k seja um inteiro não negativo. Seja R_k a relação no conjunto S dos estados de M, tal que sR_kt se e somente se para toda seqüência de entrada x com $l(x) \leq k$ [em que $l(x)$ é o comprimento de x, como sempre], $f(s, x)$ e $f(t, x)$ são ambos estados finais ou ambos estados não finais. Além disso, seja R_* a relação no conjunto dos estados de M, tal que sR_*t se e somente se para toda a seqüência de entrada x, independentemente do comprimento, $f(s, x)$ e $f(t, x)$ são ambos estados finais ou ambos estados não finais.

***58.** **a)** Mostre que para todo inteiro não negativo k, R_k é uma relação de equivalência em S. Dizemos que dois estados s e t são ***k*-equivalentes** se sR_kt.

b) Mostre que R_* é uma relação de equivalência em S. Dizemos que dois estados s e t são **$*$-equivalentes** se sR_*t.

c) Mostre que se s e t forem dois estados k-equivalentes de M, em que k é um inteiro positivo, então s e k também são $(k - 1)$-equivalentes.

d) Mostre que as classes de equivalência de R_k são um refinamento das classes de equivalência de R_{k-1} se k for um inteiro positivo. (Um refinamento de uma partição de um conjunto é definido no preâmbulo do Exercício 49 da Seção 8.5.)

e) Mostre que se s e t forem k-equivalentes para todo inteiro não negativo k, então eles são $*$-equivalentes.

f) Mostre que todos os estados em uma classe de R_*-equivalência são estados finais ou todos são estados não finais.

g) Mostre que se s e t forem R_*-equivalentes, então $f(s, a)$ e $f(t, a)$ também são R_*-equivalentes, para todo $a \in I$.

***59.** Mostre que existe um inteiro não negativo n tal que o conjunto de classes de n-equivalência de estados de M é o mesmo que o conjunto das classes de $(n + 1)$-equivalência de estados de M. A seguir, mostre que para este inteiro n, o conjunto das classes de n-equivalência de estados de M é igual ao conjunto das classes de $*$-equivalência de estados de M.

O **autômato quociente** $\overline{M}$ do autômato finito determinístico $M = (S, I, f, s_0, F)$ é o autômato finito $(\overline{S}, \overline{I}, \overline{f}, [s_0]_{R*}, \overline{F})$, em que o conjunto de estados $\overline{S}$ é o conjunto das classes de $*$-equivalência de S, a função de transição $\overline{f}$ é definida por $\overline{f}([s]_{R*}, a) = [f(s, a)]_{R*}$ para todos os estados $[s]$ de $\overline{M}$ e símbolos de entrada $a \in I$, e $\overline{F}$ é o conjunto que consiste nas classes de R_*-equivalência de estados finais de M.

***60.** **a)** Mostre que s e t são 0-equivalentes se e somente se ou ambos s e t são estados finais ou nem s nem t são estados finais. Conclua que cada estado final de $\overline{M}$, que é uma classe de R_*-equivalência, contém apenas estados finais de M.

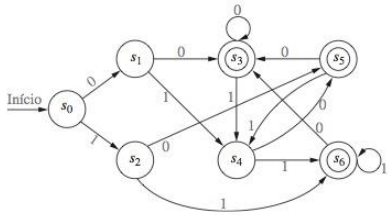
b) Mostre que se k for um inteiro positivo, então s e t são k-equivalentes se e somente se s e t forem $(k - 1)$-equivalentes e para todo símbolo de entrada $a \in I$, $f(s, a)$ e $f(t, a)$ são $(k - 1)$ equivalentes. Conclua que a função de transição $\overline{f}$ está bem definida.

c) Descreva um procedimento que possa ser usado para construir o autômato quociente de um autômato finito M.

***61.** **a)** Mostre que se M for um autômato finito, então o autômato quociente $\overline{M}$ reconhece a mesma linguagem que M.

b) Mostre que se M for um autômato finito com a propriedade que para todo estado s de M existe uma seqüência $x \in I^*$ tal que $f(s_0, x) = s$, então o autômato quociente $\overline{M}$ tem o número mínimo de estados entre todos os autômatos finitos equivalentes a M.

62. Responda a estas perguntas sobre o autômato finito M mostrado aqui.



a) Encontre as classes de k-equivalência de M para $k = 0$, 1, 2 e 3. Encontre também as classes de $*$-equivalência de M.

b) Construa o autômato quociente $\overline{M}$ de M.

12.4 Reconhecimento de Linguagem

Introdução

Vimos que os autômatos finitos podem ser usados como reconhecedores de linguagens. Quais conjuntos podem ser reconhecidos por essas máquinas? Embora este pareça ser um problema extremamente difícil, existe uma caracterização simples dos conjuntos que podem ser reconhecidos por autômatos finitos. Este problema foi primeiramente resolvido em 1956 pelo matemático norte-americano Stephen Kleene. Ele mostrou que existe um autômato finito que reconhece um conjunto se e somente se este conjunto puder ser construído a partir do conjunto nulo, a seqüência vazia, e seqüências unitárias, tomando concatenações, uniões e fechos de Kleene, em ordem arbitrária. Os conjuntos que podem ser construídos dessa forma são chamados de **conjuntos regulares**.

As gramáticas regulares foram definidas na Seção 12.1. Por causa da terminologia usada, não é surpresa que exista uma conexão entre conjuntos regulares, que são conjuntos reconhecidos por autômatos finitos, e gramáticas regulares. Em particular, um conjunto é regular se e somente se ele for gerado por uma gramática regular.

Finalmente, existem conjuntos que não podem ser reconhecidos por nenhum autômato finito. Daremos um exemplo de um desses conjuntos. Discutiremos brevemente modelos mais poderosos de computação, tal como os autômatos de pilha e máquinas de Turing, no final desta seção.

Conjuntos Regulares

Os conjuntos regulares são aqueles que podem ser formados usando-se as operações de concatenação, união e fecho de Kleene em ordem arbitrária, começando com o conjunto vazio, a seqüência vazia e conjuntos unitários. Veremos que os conjuntos regulares são aqueles que podem ser reconhecidos usando-se um autômato finito. Para definir conjuntos regulares, em primeiro lugar precisamos definir expressões regulares.

DEFINIÇÃO 1 As *expressões regulares* em um conjunto I são definidas recursivamente por:
o símbolo Ø é uma expressão regular;
o símbolo λ é uma expressão regular;
o símbolo $\boldsymbol{x}$ é uma expressão regular sempre que $x \in \boldsymbol{I}$;
os símbolos **(AB)** e **(A ∪ B)** e **A*** são expressões regulares sempre que **A** e **B** forem expressões regulares.

Cada expressão regular representa um conjunto especificado por estas regras:

Ø representa o conjunto vazio, ou seja, o conjunto sem nenhuma seqüência;
λ representa o conjunto $\{\lambda\}$, que é o conjunto que contém a seqüência vazia;
x representa o conjunto $\{x\}$, que contém a seqüência com um único símbolo x;
(AB) representa a concatenação dos conjuntos representados por **A** e por **B**;
(A ∪ B) representa a união dos conjuntos representados por **A** e por **B**;
A* representa o fecho de Kleene do conjunto representado por **A**.

Os conjuntos representados por expressões regulares são chamados de **conjuntos regulares**. De agora em diante, as expressões regulares serão usadas para descrever conjuntos regulares, de modo que quando nos referirmos ao conjunto regular **A**, o que queremos dizer é o conjunto regular representado pela expressão regular **A**. O Exemplo 1 mostra como as expressões regulares são usadas para especificar conjuntos regulares.

EXEMPLO 1 Quais são as seqüências no conjunto regular especificado pelas expressões regulares **10***, **(10)***, **0 ∪ 01**, **0(0 ∪ 1)*** e **(0*1)***?

Solução: Os conjuntos regulares especificados por essas expressões estão dados na Tabela 1, como o leitor deve verificar. ◀

Encontrar uma expressão regular que especifique um conjunto dado pode ser bastante complicado, como ilustra o Exemplo 2.

EXEMPLO 2 Encontre uma expressão regular que especifique cada um destes conjuntos:

(a) o conjunto das seqüências de bits com comprimento par
(b) o conjunto das seqüências de bits que terminam com um 0 e não contenham 11
(c) o conjunto das seqüências de bits que contenham um número ímpar de 0s

Solução: (a) Para construir uma expressão regular para o conjunto de seqüências de bits com comprimento par, usamos o fato que essa seqüência pode ser obtida concatenando-se zero ou mais seqüências, cada uma consistindo em dois bits. O conjunto de seqüências de dois bits é especificado pela expressão regular **(00 ∪ 01 ∪ 10 ∪ 11)**. Conseqüentemente, o conjunto das seqüências com comprimento par é especificado por **(00 ∪ 01 ∪ 10 ∪ 11)***.

TABELA 1

Expressão	*Seqüências*
10*	um 1 seguido por qualquer número de 0s (incluindo nenhum zero)
(10)*	qualquer número de cópias de 10 (incluindo a seqüência nula)
0 ∪ 01	a seqüência 0 ou a seqüência 01
0(0 ∪ 1)*	qualquer seqüência que comece com 0
(0*1)*	qualquer seqüência que não termine com 0

(b) Uma seqüência de bits que termine com um 0 e que não contenha 11 deve ser a concatenação de uma ou mais seqüências em que cada seqüência é um 0 ou um 10. (Para perceber isso, observe que essa seqüência de bits deve consistir em 0 bits ou em 1 bits seguido por um 0; a seqüência não pode terminar com um único 1, pois sabemos que ela termina com um 0.) Segue que a expressão regular $(\mathbf{0} \cup \mathbf{10})^* (\mathbf{0} \cup \mathbf{10})$ especifica o conjunto das seqüências de bits que não contêm 11 e terminam com um 0. [Observe que o conjunto especificado por $(\mathbf{0} \cup \mathbf{10})^*$ inclui a seqüência vazia, que não está neste conjunto, pois a seqüência vazia não termina com um 0.]

(c) Uma seqüência de bits com um número ímpar de 0s deve conter pelo menos um 0, o que nos diz que ela começa com zero ou mais 1s, seguido por um 0, seguido por zero ou mais 1s, ou seja, cada uma dessas seqüências de bits começa com uma seqüência da forma 1^j01^k para inteiros não negativos j e k. Como a seqüência de bits contém um número ímpar de 0s, os bits adicionais depois deste bloco inicial podem ser divididos em blocos, cada um começando com um 0 e contendo mais um 0. Cada um desses blocos é da forma 01^p01^q, em que p e q são inteiros não negativos. Conseqüentemente, a expressão regular **1*01*(01*01*)*** especifica o conjunto de seqüências de bits com um número ímpar de 0s. ◀

Teorema de Kleene

Em 1956, Kleene demonstrou que conjuntos regulares são os conjuntos que são reconhecidos por autômatos finitos. Conseqüentemente, este importante resultado é chamado de Teorema de Kleene.

TEOREMA 1

TEOREMA DE KLEENE Um conjunto é regular se e somente se ele for reconhecido por um autômato finito.

O Teorema de Kleene é um dos resultados centrais na teoria dos autômatos. Demonstraremos a parte *somente se* deste teorema, ou seja, que todo conjunto regular é reconhecido por um autômato finito. A demonstração da recíproca, que um conjunto reconhecido por um autômato finito é regular, é deixada como exercício para o leitor.

Demonstração: Lembre que um conjunto regular é definido em termos de expressões regulares, as quais são definidas recursivamente. Podemos demonstrar que todo conjunto regular é reconhecido por um autômato finito se pudermos fazer o seguinte.

1. Mostrar que $\emptyset$ é reconhecido por um autômato finito.
2. Mostrar que $\{\lambda\}$ é reconhecido por um autômato finito.
3. Mostrar que $\{a\}$ é reconhecido por um autômato finito sempre que a for um símbolo em I.
4. Mostrar que AB é reconhecido por um autômato finito sempre que ambos A e B forem reconhecidos por ele.
5. Mostrar que $A \cup B$ é reconhecido por um autômato finito sempre que ambos A e B forem reconhecidos por ele.
6. Mostrar que A^* é reconhecido por um autômato finito sempre que A for reconhecido por ele.

Consideramos agora cada uma dessas tarefas. Primeiro, mostramos que $\emptyset$ é reconhecido por um autômato finito não determinístico. Para fazer isso, tudo o que precisamos é de um autômato sem nenhum estado finito. Esse autômato é mostrado na Figura 1(a).

Segundo, mostramos que $\{\lambda\}$ é reconhecido por um autômato finito. Para fazer isso, tudo o que precisamos é de um autômato que reconheça λ, a seqüência nula, mas nenhuma outra seqüência. Isso pode ser realizado fazendo o estado inicial s_0 um estado final e sem ter transições, de modo que nenhuma outra seqüência leva s_0 para um estado final. O autômato não determinístico na Figura 1(b) mostra uma destas máquinas.

Terceiro, mostramos que $\{a\}$ é reconhecido por um autômato finito não determinístico. Para fazer isso, podemos usar uma máquina com um estado inicial s_0 e um estado final s_1. Temos uma

FIGURA 1 Autômatos Finitos Não Determinísticos que Reconhecem Alguns Conjuntos Básicos.

transição de s_0 para s_1 quando a entrada for a e não temos nenhuma outra transição. A única seqüência reconhecida por esta máquina é a. Esta máquina é apresentada na Figura 1(c).

A seguir, mostramos que AB e $A \cup B$ podem ser reconhecidos por autômatos finitos se A e B forem linguagens reconhecidas por autômatos finitos. Suponha que A seja reconhecido por $M_A = (S_A, I, f_A, s_A, F_A)$ e B seja reconhecido por $M_B = (S_B, I, f_B, s_B, F_B)$.

Começamos com a construção de uma máquina de estados finitos $M_{AB} = (S_{AB}, I, f_{AB}, s_{AB}, F_{AB})$ que reconhece AB, a concatenação de A e B. Construímos essa máquina combinando as máquinas para A e B em séries, de modo que uma seqüência em A leva a máquina combinada de s_A, o estado inicial de M_A, para s_B, o estado inicial de M_B. Uma seqüência em B deve levar a máquina combinada de s_B para um estado final da máquina combinada. Conseqüentemente, fazemos a seguinte construção. Considere S_{AB} como $S_A \cup S_B$. O estado inicial s_{AB} é o mesmo que s_A. O conjunto dos estados finais, F_{AB}, é o conjunto dos estados finais de M_B com s_{AB} incluído se e somente se $\lambda \in A \cap B$. As transições em M_{AB} incluem todas as transições em M_A e em M_B, bem como algumas novas transições. Para toda transição em M_A que leve a um estado final, formamos uma transição em M_{AB} do mesmo estado para s_B, com a mesma entrada. Dessa maneira, uma seqüência em A leva M_{AB} de s_{AB} para s_B, e, então, uma seqüência em B leva s_B para um estado final de M_{AB}. Além disso, para toda transição a partir de s_B, formamos uma transição em M_{AB} de s_{AB} para o mesmo estado. A Figura 2(a) contém uma ilustração dessa construção.

Agora construímos uma máquina $M_{A \cup B} = (S_{A \cup B}, I, f_{A \cup B}, s_{A \cup B}, F_{A \cup B})$ que reconhece $A \cup B$. Este autômato pode ser construído combinando-se M_A e M_B em paralelo, usando-se um novo estado inicial que tem as transições que tanto s_A quanto s_B têm. Seja $S_{A \cup B} = S_A \cup S_B \cup \{s_{A \cup B}\}$, em que $s_{A \cup B}$ é um novo estado que é o estado inicial de $M_{A \cup B}$. Defina o conjunto dos estados finais $F_{A \cup B}$ como $F_A \cup F_B \cup \{s_{A \cup B}\}$ se $\lambda \in A \cup B$, e $F_A \cup F_B$, caso contrário. As transições em $M_{A \cup B}$ incluem todas aquelas em M_A e em M_B. Além disso, para cada transição a partir de s_A para um estado s com entrada i, incluímos uma transição de $s_{A \cup B}$ para s com entrada i, e para cada transição de s_B para um estado s com entrada i, incluímos uma transição de $s_{A \cup B}$ para s com entrada i. Dessa maneira, uma seqüência em A leva de $s_{A \cup B}$ para um estado final na nova máquina, e uma seqüência em B leva de $s_{A \cup B}$ para um estado final na nova máquina. A Figura 2(b) ilustra a construção de $M_{A \cup B}$.

Finalmente, construímos $M_A* = (S_{A^*}, I, f_{A^*}, s_{A^*}, F_{A^*})$, uma máquina que reconhece A^*, o fecho de Kleene de A. Inclua em S_{A^*} todos os estados em S_A e um estado adicional s_{A^*}, o qual é o estado inicial da nova máquina. O conjunto de estados finais F_{A^*} inclui todos os estados em F_A, bem como o estado inicial s_{A^*}, pois λ deve ser reconhecida. Para reconhecer concatenações de um número arbitrário de seqüências de A, incluímos todas as transições em M_A, bem como todas as transições a partir de s_{A^*} que correspondam às transições a partir de s_A, e as transições a partir de cada estado final que correspondam às transições a partir de s_A. Com este conjunto de transições, uma seqüência formada por concatenações de seqüências de A levarão s_{A^*} para um estado final quando a primeira seqüência em A tiver sido lida, retornando para um estado final quando a segunda seqüência em A tiver sido lida, e assim por diante. A Figura 2(c) ilustra a construção que usamos. ◁

Um autômato finito não determinístico pode ser construído para qualquer conjunto regular, usando o procedimento descrito nesta demonstração. Ilustramos como isso é feito com o Exemplo 3.

EXEMPLO 3 Construa um autômato finito não determinístico que reconheça o conjunto regular **1*** ∪ **01**.

Solução: Começamos com a construção de uma máquina que reconheça **1***. Isso é feito usando a máquina que reconhece **1** e a seguir usando a construção para M_{A^*} descrita na demonstração.

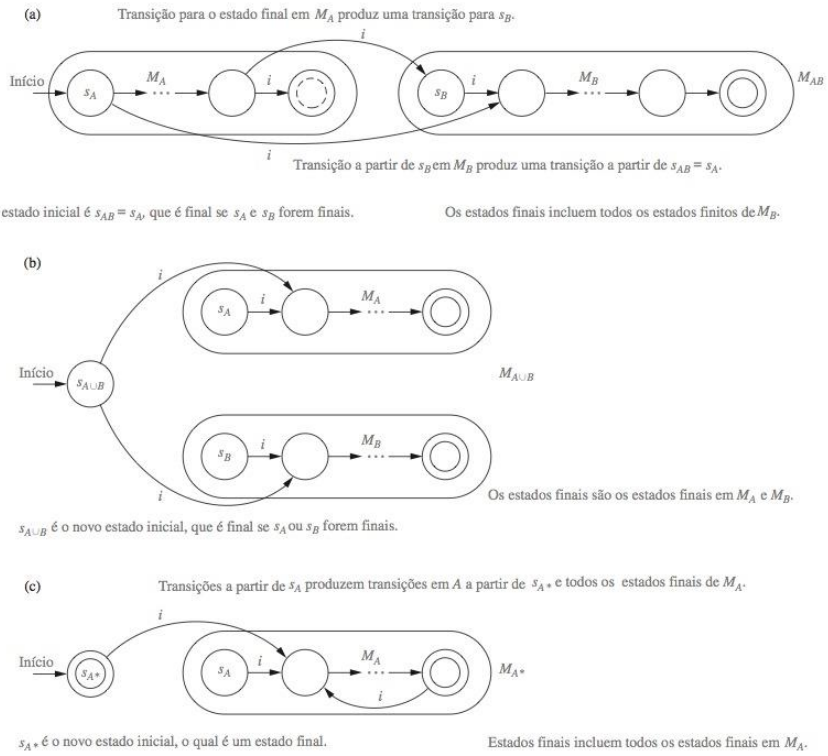


FIGURA 2 Construção de Autômatos para Reconhecer Concatenações, Uniões e Fechos de Kleene.

A seguir, construímos uma máquina que reconheça **01**, usando máquinas que reconheçam **0** e **1** e a construção na demonstração para M_{AB}. Finalmente, usando a construção na demonstração para $M_{A\cup B}$, construímos a máquina para **1*** ∪ **01**. Os autômatos finitos usados nesta construção são mostrados na Figura 3. Os estados nestas máquinas sucessivas foram rotulados com diferentes subscritos, mesmo quando um estado é formado a partir de um usado previamente em uma outra máquina. Observe que a construção dada aqui não produz a máquina mais simples que reconhece **1*** ∪ **01**. Uma máquina muito mais simples que reconhece este conjunto é mostrada na Figura 3(b). ◀

Conjuntos Regulares e Gramáticas Regulares

Na Seção 12.1, introduzimos gramáticas de estrutura de frase e definimos tipos diferentes de gramáticas. Em particular, definimos gramáticas regulares, ou de tipo 3, as quais são gramáticas da forma $G = (V, T, S, P)$, em que cada regra de produção é da forma $S \to \lambda$, $A \to a$ ou $A \to aB$,

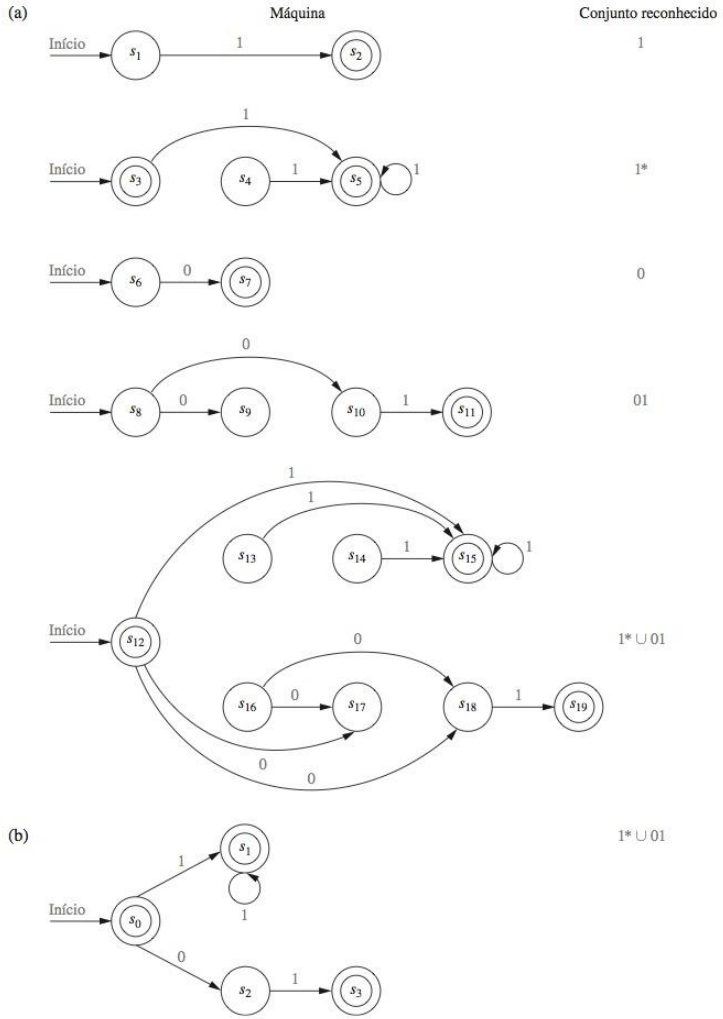


FIGURA 3 Autômatos Finitos Não Determinísticos que Reconhecem 1* ∪ 01.

em que a é um símbolo terminal e A e B são símbolos não terminais. Como a terminologia sugere, existe uma ligação íntima entre gramáticas regulares e conjuntos regulares.

TEOREMA 2 Um conjunto é gerado por uma gramática regular se e somente se ele for um conjunto regular.

Demonstração: Primeiro mostramos que um conjunto gerado por uma gramática regular é um conjunto regular. Suponha que $G = (V, T, S, P)$ seja uma gramática regular que gere o conjunto $L(G)$. Para mostrar que $L(G)$ é regular, construiremos uma máquina de estados finitos não determinística $M = (S, I, f, s_0, F)$ que reconheça $L(G)$. Faça com que S, o conjunto dos estados, contenha um estado s_A para cada símbolo não terminal A de G e um estado adicional s_F, que é um estado final. O estado inicial s_0 é o estado formado a partir do símbolo inicial S. As transições de M são formadas a partir das regras de produção de G da maneira a seguir. Uma transição de s_A para s_F com entrada de a é incluída se $A \rightarrow a$ for uma regra de produção, e uma transição de s_A para s_B com a entrada de a é incluída se $A \rightarrow aB$ for uma regra de produção. O conjunto dos estados finais inclui s_F e também inclui s_0 se $S \rightarrow \lambda$ for uma regra de produção em G. Não é difícil mostrar que a linguagem reconhecida por M coincide com a linguagem gerada pela gramática G, ou seja, $L(M) = L(G)$. Isso pode ser feito determinando-se as palavras que levam a um estado final. Os detalhes são deixados como exercício para o leitor. ◁

Antes de apresentar a demonstração da recíproca, ilustraremos como é construída uma máquina não determinística que reconheça o mesmo conjunto que uma gramática regular.

EXEMPLO 4 Construa um autômato finito não determinístico que reconheça a linguagem gerada pela gramática regular $G = (V, T, S, P)$, em que $V = \{0, 1, A, S\}$, $T = \{0, 1\}$ e as regras de produção em P são $S \rightarrow 1A$, $S \rightarrow 0$, $S \rightarrow \lambda$, $A \rightarrow 0A$, $A \rightarrow 1A$ e $A \rightarrow 1$.

Solução: O diagrama de estados para um autômato finito não determinístico que reconhece $L(G)$ é mostrado na Figura 4. Este autômato é construído seguindo o procedimento descrito na demonstração. Neste autômato, s_0 é o estado correspondente a S, s_1 é o estado correspondente a A e s_2 é o estado final. ◀

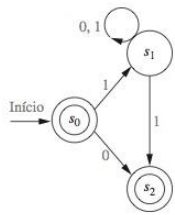


FIGURA 4
Autômato Finito Não Determinístico que Reconhece $L(G)$.

Completaremos agora a demonstração do Teorema 2.

Demonstração: Mostramos agora que, se um conjunto for regular, então existe uma gramática regular que o gera. Suponha que M seja uma máquina de estados finitos que reconheça este conjunto com a propriedade que s_0, o estado inicial de M, nunca é o estado seguinte de uma transição. (Podemos encontrar uma máquina assim no Exercício 20.) A gramática $G = (V, T, S, P)$ é definida como a seguir. O conjunto V de símbolos de G é formado, associando-se um símbolo a cada estado de S e a cada símbolo de entrada em I. O conjunto T de símbolos terminais de G é o conjunto dos símbolos de G formados a partir de símbolos de entrada em I. O símbolo inicial S é o símbolo formado a partir do estado inicial s_0. O conjunto P das regras de produção em G é formado a partir das transições em M. Em particular, se o estado s vai para um estado final com a entrada a, então a regra de produção $A_s \rightarrow a$ é incluída em P, em que A_s é o símbolo não terminal formado a partir do estado s. Se o estado s for para o estado t com a entrada a, então a regra de produção $A_s \rightarrow aA_t$ é incluída em P. A regra de produção $S \rightarrow \lambda$ é incluída em P se e somente se $\lambda \in L(M)$. Como as regras de produção de G correspondem às transições de M e as regras de produção que levam aos terminais correspondem a transições para estados finais, não é difícil mostrar que $L(G) = L(M)$. Deixamos os detalhes como exercício para o leitor. ◁

O Exemplo 5 ilustra a construção usada para produzir, a partir de um autômato, uma gramática que gera a linguagem reconhecida por este autômato.

EXEMPLO 5 Encontre uma gramática regular que gere o conjunto regular reconhecido pelo autômato finito mostrado na Figura 5.

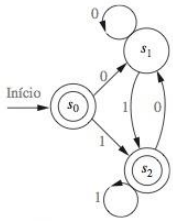


FIGURA 5
Autômato Finito.

Solução: A gramática $G = (V, T, S, P)$ gera o conjunto reconhecido por este autômato, em que $V = (S, A, B, 0, 1)$; em que os símbolos S, A e B correspondem aos estados s_0, s_1 e s_2, respectivamente; $T = \{0, 1\}$; S é o símbolo inicial; e as regras de produção são $S \to 0A$, $S \to 1B$, $S \to 1$, $S \to \lambda$, $A \to 0A$, $A \to 1B$, $A \to 1$, $B \to 0A$, $B \to 1B$ e $B \to 1$. ◀

Um Conjunto que Não É Reconhecido por um Autômato Finito

Vimos que um conjunto é reconhecido por um autômato finito se e somente se ele é regular. Mostraremos agora que existem conjuntos que não são regulares, descrevendo um desses conjuntos. A técnica usada para mostrar que este conjunto não é regular ilustra um método importante para demonstrar que determinados conjuntos não são regulares.

EXEMPLO 6 Mostre que o conjunto $\{0^n1^n \mid n = 0, 1, 2, \ldots\}$, formado de todas as seqüências que consistem em um bloco de 0s seguido por um bloco com o mesmo número de 1s, não é regular.

Solução: Suponha que este conjunto seja regular. Então, existiria um autômato finito não determinístico $M = (S, I, f, s_0, F)$ que o reconheceria. Seja N o número de estados nesta máquina, ou seja, $N = |S|$. Como M reconhece todas as seqüências formadas por determinado número de 0s seguidos pelo mesmo número de 1s, M deve reconhecer 0^N1^N. Seja $s_0, s_1, s_2, \ldots, s_{2N}$ a seqüência de estados que é obtida começando em s_0 e usando os símbolos de 0^N1^N como entrada, de modo que $s_1 = f(s_0, 0)$, $s_2 = f(s_1, 0), \ldots, s_N = f(s_{N-1}, 0)$, $s_{N+1} = f(s_N, 1), \ldots, s_{2N} = f(s_{2N-1}, 1)$. Observe que s_{2N} é um estado final.

Como existem apenas N estados, o princípio da casa do pombo mostra que pelo menos dois dos primeiros $N + 1$ estados, que são $s_0, \ldots, s_N$, devem ser os mesmos. Digamos que s_i e s_j sejam dois estados idênticos, com $0 \le i < j \le N$. Isso significa que $f(s_i, 0^t) = s_j$, em que $t = j - i$. Segue que existe um laço que leva de s_i de volta para ele mesmo, obtido usando a entrada 0 um total de t vezes, no diagrama de estado mostrado na Figura 6.

Considere agora a seqüência de entrada $0^N0^t1^N = 0^{N+t}1^N$. Existem t 0s consecutivos a mais no início deste bloco do que 1s consecutivos a seguir. Como esta seqüência não é da forma 0^n1^n (pois tem mais 0s do que 1s), ela não é reconhecida por M. Conseqüentemente, $f(s_0, 0^{N+t}1^N)$ não pode ser um estado final. Entretanto, quando usamos a seqüência $0^{N+t}1^N$ como entrada, acabamos com o mesmo estado que antes, ou seja, s_{2N}. A razão para isso é que os t 0s extras nesta seqüência nos levam ao redor do laço de s_i de volta para ele mesmo mais uma vez, como mostrado

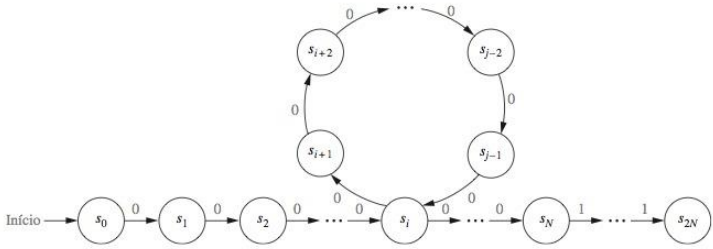


FIGURA 6 O Caminho Produzido por 0^N1^N.

na Figura 6. Então, o resto da seqüência nos leva exatamente para o mesmo estado que antes. Esta contradição mostra que $\{0^n1^n \mid n = 1, 2, \ldots\}$ não é regular. ◀

Tipos de Máquinas mais Poderosas

Autômatos finitos são incapazes de efetuar muitos cálculos. A principal limitação dessas máquinas é sua quantidade finita de memória. Isso as impede de reconhecer linguagens que não são regulares, tais como $\{0^n1^n \mid n = 0, 1, 2, \ldots\}$. Como um conjunto é regular se e somente se ele é a linguagem gerada por uma gramática regular, o Exemplo 5 mostra que não existe nenhuma gramática regular que gere o conjunto $\{0^n1^n \mid n = 0, 1, 2, \ldots\}$. Entretanto, existe uma gramática livre de contexto que reconhece esse conjunto. Essa gramática foi mostrada no Exemplo 5 da Seção 12.1.

Por causa das limitações das máquinas de estados finitos, é necessário usar outros modelos computacionais mais poderosos. Um desses modelos é o **autômato de pilha**. Um autômato de pilha inclui tudo em um autômato finito, bem como uma pilha, que fornece memória ilimitada. Os símbolos podem ser colocados no topo ou tirados do topo da pilha. Um conjunto é reconhecido de uma entre duas maneiras em um autômato de pilhas. Primeiro, um conjunto é reconhecido se o conjunto consiste em todas as seqüências que produzem uma pilha vazia quando são usadas como entrada. Segundo, um conjunto é reconhecido se ele consistir em todas as seqüências que levam a um estado final quando usadas como entrada. Pode ser mostrado que um conjunto é reconhecido por um autômato de pilha se e somente se ele for a linguagem gerada por uma gramática livre de contexto.

Entretanto, existem conjuntos que não podem ser expressos como a linguagem gerada por uma gramática livre de contexto. Um desses conjuntos é $\{0^n1^n2^n \mid n = 0, 1, 2, \ldots\}$. Indicaremos porque este conjunto não pode ser reconhecido por um autômato de pilha, mas não daremos uma demonstração, pois não desenvolvemos as ferramentas necessárias. (Entretanto, um método de demonstração é dado no Exercício 28 dos exercícios complementares no final deste capítulo.) A pilha pode ser usada para mostrar que uma seqüência começa com uma seqüência de 0s seguida por um número igual de 1s que coloca um símbolo na pilha para cada 0 (enquanto apenas 0s forem lidos) e que remove um desses símbolos para cada 1 (enquanto forem lidos apenas 1s a seguir dos 0s). Contudo, uma vez que isso for realizado, a pilha está vazia e não há maneira de determinar que exista o mesmo número de 2s do que de 0s na seqüência.

Existem outras máquinas, chamadas de **autômatos linearmente limitados**, mais poderosos que os autômatos de pilhas, que podem reconhecer conjuntos tais como $\{0^n1^n2^n \mid n = 0, 1, 2, \ldots\}$. Em particular, os autômatos linearmente limitados podem reconhecer linguagens sensíveis ao contexto. Entretanto, essas máquinas não podem reconhecer todas as linguagens geradas por gramáticas de estrutura de frase. Para evitar as limitações de tipos especiais de máquinas, é usado o modelo conhecido como uma **máquina de Turing**, assim chamada em homenagem ao matemático inglês Alan Turing. Uma máquina de Turing é formada de tudo que é incluído em uma máquina finita junto com uma fita, que é infinita em ambas as direções. Uma máquina de Turing tem capacidades de leitura e escrita na fita, e pode se mover para frente e para trás nessa fita. As máquinas de Turing podem reconhecer todas as linguagens geradas por gramáticas de estrutura de frase. Além disso, podem modelar todos os cálculos que podem ser executados em uma máquina computacional. Por causa de seu poder, as máquinas de Turing são amplamente estudadas em ciência da computação teórica. Discutiremos sobre essas máquinas brevemente na Seção 12.5.

Exercícios

1. Descreva em palavras as seqüências em cada um destes conjuntos regulares.

a) 1*0 **b)** 1*00*
c) 111 ∪ 001 **d)** (1 ∪ 00)*
e) (00*1)* **f)** (0 ∪ 1)(0 ∪ 1)*00

2. Descreva em palavras as seqüências em cada um destes conjuntos regulares.

a) 001* **b)** (01)*
c) 01 ∪ 001* **d)** 0(11 ∪ 0)*
e) (101*)* **f)** (0* ∪ 1)11

3. Determine se 0101 pertence a cada um destes conjuntos regulares.
a) **01*0** **b)** **0(11)*(01)***
c) **0(10)* 1*** **d)** **0*10(0 ∪ 1)**
e) **(01)*(11)*** **f)** **0*(10 ∪ 11)***
g) **0*(10)*11** **h)** **01(01 ∪ 0)1***

4. Determine se 1011 pertence a cada um destes conjuntos regulares.
a) **10*1*** **b)** **0*(10 ∪ 11)***
c) **1(01)* 1*** **d)** **1*01(0 ∪ 1)**
e) **(10)*(11)*** **f)** **1(00)*(11)***
g) **(10)*1011** **h)** **(1 ∪ 00)(01 ∪ 0)1***

5. Expresse cada um destes conjuntos usando uma expressão regular.
a) o conjunto que consiste nas seqüências 0, 11 e 010
b) o conjunto de seqüências de três 0s seguidos por dois ou mais 0s
c) o conjunto das seqüências de comprimento ímpar
d) o conjunto das seqüências que contêm exatamente um 1
e) o conjunto das seqüências que terminam com 1 e não contêm 000

6. Expresse cada um destes conjuntos usando uma expressão regular.
a) o conjunto que contém todas as seqüências com 0, 1 ou 2 bits
b) o conjunto de seqüências de dois 0s, seguidos por zero ou mais 1s e terminando com um 0
c) o conjunto das seqüências com todo 1 seguido por dois 0s
d) o conjunto das seqüências que terminam em 00 e não contêm 11
e) o conjunto das seqüências que contêm um número par de 1s

7. Expresse cada um destes conjuntos usando uma expressão regular.
a) o conjunto das seqüências de um ou mais 0s seguidos por um 1
b) o conjunto de seqüências de dois ou mais símbolos seguidos por três ou mais 0s
c) o conjunto das seqüências sem nenhum 1 que precede um 0 ou sem nenhum 0 que precede um 1
d) o conjunto das seqüências que contêm uma seqüência de 1s tal que o número de 1s é igual a 2 módulo 3, seguido por um número par de 0s

8. Construa um autômato finito determinístico que reconheça cada um destes conjuntos de I^*, em que I é um alfabeto.
a) $\emptyset$ **b)** $\{\lambda\}$ **c)** $\{a\}$, em que $a \in I$

9. Construa um autômato finito não determinístico que reconheça cada um dos conjuntos do Exercício 8.

10. Construa um autômato finito não determinístico que reconheça cada um dos seguintes conjuntos
a) $\{\lambda, 0\}$. **b)** $\{0, 11\}$. **c)** $\{0, 11, 000\}$.

***11.** Mostre que se A for um conjunto regular, então A^R, o conjunto de todos os reversos das seqüências em A, também é regular.

12. Usando as construções descritas na demonstração do Teorema de Kleene, encontre autômatos finitos não determinísticos que reconheçam cada um destes conjuntos.
a) **01*** **b)** **(0 ∪ 1)1*** **c)** **00(1* ∪ 10)**

13. Usando as construções descritas na demonstração do Teorema de Kleene, encontre autômatos finitos não determinísticos que reconheçam cada um destes conjuntos.
a) **0*1*** **b)** **(0 ∪ 11)*** **c)** **01* ∪ 00*1**

14. Construa um autômato finito não determinístico que reconheça a linguagem gerada pela gramática regular $G = (V, T, S, P)$, em que $V = \{0, 1, S, A, B\}$, $T = \{0, 1\}$, S é o símbolo inicial e o conjunto das regras de produção é
a) $S \to 0A$, $S \to 1B$, $A \to 0$, $B \to 0$.
b) $S \to 1A$, $S \to 0$, $S \to \lambda$, $A \to 0B$, $B \to 1B$, $B \to 1$.
c) $S \to 1B$, $S \to 0$, $A \to 1A$, $A \to 0B$, $A \to 1$, $A \to 0$, $B \to 1$.

Nos exercícios 15 a 17, construa uma gramática regular $G = (V, T, S, P)$ que gere a linguagem reconhecida pela máquina de estados finitos dada.

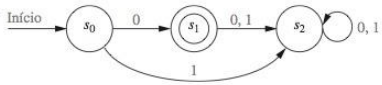
Links

ALAN MATHISON TURING (1912–1954) Alan Turing nasceu em Londres, embora tenha sido concebido na Índia, onde seu pai estava a serviço da Indian Civil Service. Quando menino, era fascinado por química, tendo feito uma grande variedade de experiências, e por maquinaria. Turing freqüentou Sherborne, um internato inglês. Em 1931, ganhou uma bolsa para o King's College, em Cambridge. Depois de terminar sua dissertação, que incluiu uma redescoberta do Teorema do Limite Central, um teorema famoso em estatística, ele foi eleito membro de sua faculdade. Em 1935, Turing ficou fascinado pelo problema de decisão, um problema proposto pelo grande matemático alemão Hilbert, que perguntou se existe um método geral que pode ser aplicado a qualquer afirmação para determinar se ela é verdadeira. Turing gostava de correr (tendo mais tarde corrido em competições como um amador sério) e um dia, enquanto descansava depois de uma corrida, ele descobriu as idéias-chave necessárias para resolver o problema de decisão. Em sua solução, ele inventou o que hoje é chamado de **máquina de Turing** como o modelo mais geral de uma máquina computacional. Usando essas máquinas, ele encontrou um problema, que envolvia o que ele chamou de números computáveis, que não poderiam ser decididos usando um método geral.

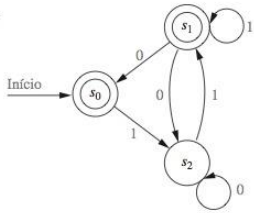
De 1936 a 1938, Turing visitou a Universidade de Princeton para trabalhar com Alonzo Church, que também tinha resolvido o problema de decisão de Hilbert. Em 1939, Turing voltou para o King's College. Entretanto, com o desencadear da segunda guerra mundial, ele ingressou no Foreign Office, fazendo criptoanálise dos códigos alemães. Sua contribuição para quebrar o código do Enigma, uma máquina codificadora mecânica alemã, desempenhou um papel importante para vencer a guerra.

Depois da guerra, Turing trabalhou no desenvolvimento dos primeiros computadores. Ele estava interessado na habilidade de as máquinas pensarem, propondo que se um computador não podia ser distinguido de uma pessoa com base em respostas escritas a perguntas, ele deveria ser considerado "pensante". Ele também estava interessado em biologia, tendo escrito sobre morfogênese, o desenvolvimento da forma nos organismos. Em 1954, Turing cometeu suicídio tomando cianeto, sem deixar uma explicação clara. Problemas legais relacionados a uma relação homossexual e tratamentos hormonais determinados pela corte para diminuir seu impulso sexual podem ter sido fatores em sua decisão sobre acabar com sua vida.

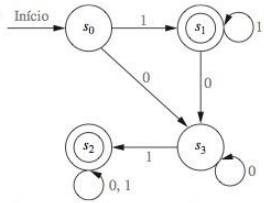
15.



16.



17.



18. Mostre que o autômato finito construído a partir de uma gramática regular na demonstração do Teorema 2 reconhece o conjunto gerado por esta gramática.

19. Mostre que a gramática regular construída a partir de um autômato finito na demonstração do Teorema 2 gera o conjunto reconhecido por este autômato.

20. Mostre que todo autômato finito não determinístico é equivalente a outro destes autômatos que têm a propriedade de que seu estado inicial nunca é revisitado.

***21.** Seja $M = (S, I, f, s_0, F)$ um autômato finito determinístico. Mostre que a linguagem reconhecida por M, $L(M)$, é infinita se e somente se existir uma palavra x reconhecida por M com $l(x) \geq |S|$.

***22.** Uma técnica importante usada para demonstrar que certos conjuntos não são regulares é o **lema do bombeamento** (*pumping lemma*). O lema do bombeamento afirma que se $M = (S, I, f, s_0, F)$ for um autômato finito determinístico e se x for uma seqüência em $L(M)$, a linguagem reconhecida por M, com $l(x) \geq |S|$, então existem seqüências u, v e w em I^*, tal que $x = uvw$, $l(uv) \leq |S|$ e $l(v) \geq 1$ e $uv^i w \in L(M)$ para $i = 0, 1, 2, \ldots$. Demonstre o lema do bombeamento. [*Dicas*: Use a mesma idéia utilizada no Exemplo 5.]

***23.** Mostre que o conjunto $\{0^{2n}1^n\}$ não é regular, usando o lema do bombeamento dado no Exercício 22.

***24.** Mostre que o conjunto $\{1^{n^2} \mid n = 0, 1, 2, \ldots\}$ não é regular, usando o lema do bombeamento dado no Exercício 22.

***25.** Mostre que o conjunto dos palíndromos sobre $\{0, 1\}$ não é regular, usando o lema do bombeamento dado no Exercício 22. [*Dica*: Considere as seqüências da forma 0^N10^N.]

****26.** Mostre que o conjunto reconhecido por um autômato finito é regular. (Esta é a parte do *se* do Teorema de Kleene.)

Suponha que L seja um subconjunto de I^*, em que I é um conjunto não vazio de símbolos. Se $x \in I^*$, fazemos $L/x = \{z \in I^* \mid xz \in L\}$. Dizemos que as seqüências $x \in I^*$ e $y \in I^*$ são **distinguíveis em relação a** L se $L/x \neq L/y$. Dizemos que uma seqüência z para a qual $xz \in L$, mas $yz \notin L$, ou $xz \notin L$, mas $yz \in L$, **distingue** x e y em relação a L. Quando $L/x = L/y$, dizemos que x e y são **indistinguíveis** em relação a L.

27. Seja L o conjunto de todas as seqüências de bits que terminam com 01. Mostre que 11 e 10 são distinguíveis em relação a L e que as seqüências 1 e 11 são indistinguíveis em relação a L.

28. Suponha que $M = (S, I, f, s_0, F)$ seja uma máquina de estados finitos determinística. Mostre que se x e y forem duas seqüências em I^* que são distinguíveis em relação a $L(M)$, então $f(s_0, x) \neq f(s_0, y)$.

***29.** Suponha que L seja um subconjunto de I^* e que, para algum inteiro positivo n, existam n seqüências em I^*, tal que todos os pares destas seqüências são distinguíveis em relação a L. Demonstre que todo autômato finito determinístico que reconheça L tem pelo menos n estados.

***30.** Seja L_n o conjunto das seqüências com pelo menos n bits nas quais o n-ésimo símbolo a partir do fim seja um 0. Use o Exercício 29 para mostrar que uma máquina de estados finitos determinística que reconheça L_n deve ter pelo menos 2^n estados.

31. Use o Exercício 29 para mostrar que a linguagem que consiste em todas as seqüências de bits que são palíndromos (isto é, seqüências que são iguais a seu próprio reverso) não é regular.

12.5 Máquinas de Turing

Introdução

Links

Os autômatos finitos estudados anteriormente neste capítulo não podem ser usados como modelos gerais de computação. Eles são limitados quanto ao que podem fazer. Por exemplo, autômatos finitos são capazes de reconhecer conjuntos regulares, mas não são capazes de reconhecer muitos conjuntos fáceis de descrever, incluindo $\{0^n1^n \mid n \geq 0\}$, os quais os computadores reconhecem usando memória. Podemos usar autômatos finitos para calcular funções relativamente simples, como a soma de dois números, mas não podemos usá-los para calcular funções que os computadores podem calcular, como o produto de dois números. Para superar estas deficiências, podemos usar um tipo de máquina mais poderoso, conhecido como máquina de Turing, em homenagem a Turing, o famoso matemático e cientista da computação que as inventou na década de 1930.

Basicamente, uma máquina de Turing consiste em uma unidade de controle, que em cada passo é um entre um número finito de estados diferentes, junto com uma fita dividida em células, que é infinita em ambos os sentidos. As máquinas de Turing têm recursos de leitura e escrita na fita à medida que a unidade de controle se move para frente e para trás ao longo desta fita, mudando os estados dependendo do símbolo lido na fita. As máquinas de Turing são mais poderosas que as máquinas de estados finitos, pois elas incluem recursos de memória que as máquinas de estados finitos não têm. Mostraremos como usar máquinas de Turing para reconhecer conjuntos, incluindo conjuntos que não podem ser reconhecidos por máquinas de estados finitos. Vamos mostrar também como calcular funções usando máquinas de Turing. As máquinas de Turing são os modelos mais gerais de computação; essencialmente, elas podem fazer tudo que um computador pode fazer. Observe que as máquinas de Turing são muito mais poderosas que os computadores reais, que têm recursos de memória finitos.

Definição de Máquinas de Turing

Damos agora a definição formal de máquinas de Turing. Posteriormente, explicaremos como esta definição formal pode ser interpretada em termos de um cabeçote de controle que pode ler e escrever símbolos em uma fita e se mover para a direita ou para a esquerda sobre esta fita.

DEFINIÇÃO 1 Uma *máquina de Turing* $T = (S, I, f, s_0)$ consiste em um conjunto finito S de estados, um alfabeto I que contém o símbolo branco (ou em branco) B, uma função parcial f de S × I para $S \times I \times \{D, E\}$ e um estado inicial s_0.

Lembre-se do preâmbulo do Exercício 73 da Seção 2.3, de que uma função parcial é definida apenas para aqueles elementos em seu domínio de definição. Isso significa que para alguns pares (estado, símbolo) a função parcial f pode não estar definida, mas para um par para o qual esteja definida, existe uma única tripla (estado, símbolo, sentido) associada a este par. Chamamos as quíntuplas correspondentes à função parcial na definição de máquina de Turing de **regras de transição** da máquina.

Para interpretar esta definição em termos de uma máquina, considere uma unidade de controle e uma fita dividida em células, infinita em ambos os sentidos, que tem apenas um número finito de símbolos não brancos nela em cada instante dado, como ilustrado na Figura 1. A ação da máquina de Turing em cada passo de sua operação depende do valor da função parcial f no estado e símbolo da fita correntes.

Em cada passo, a unidade de controle lê o símbolo corrente x na fita. Se a unidade de controle estiver no estado s e se a função parcial f estiver definida no par (s, x) com $f(s, x) = (s', x', d)$, a unidade de controle

1. entra no estado s',
2. escreve o símbolo x' na célula corrente, apagando x, e
3. move-se para a direita uma célula se $d = D$ ou move-se para a esquerda uma célula se $d = E$.

Escrevemos este passo como a quíntupla (s, x, s', x', d). Se a função parcial f não estiver definida para o par (s, x), então a máquina de Turing T vai *parar*.

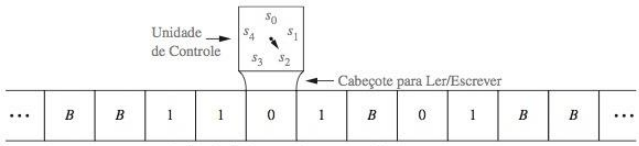


FIGURA 1 **Representação de uma Máquina de Turing.**

Uma maneira comum de definir uma máquina de Turing é especificar um conjunto de quíntuplas da forma (s, x, s', x', d). O conjunto de estados e o alfabeto de entrada estão implicitamente definidos quando essa definição é usada.

No início de sua operação, é suposto que uma máquina de Turing esteja em seu estado inicial s_0 e que esteja posicionada no símbolo não branco mais à esquerda na fita. Se a fita estiver toda em branco, o cabeçote de controle pode ser posicionado em qualquer célula. Chamaremos o posicionamento do cabeçote de controle no símbolo não branco da fita mais à esquerda de *posição inicial* da máquina.

O Exemplo 1 ilustra como funciona uma máquina de Turing.

EXEMPLO 1 Qual é a fita final quando a máquina de Turing T definida pelas sete quíntuplas $(s_0, 0, s_0, 0, R)$, $(s_0, 1, s_1, 1, R)$, (s_0, B, s_3, B, R), $(s_1, 0, s_0, 0, R)$, $(s_1, 1, s_2, 0, L)$, (s_1, B, s_3, B, R) e $(s_2, 1, s_3, 0, R)$ é executada na fita mostrada na Figura 2(a)?

Exemplos Extras

Solução: Começamos a operação com T no estado s_0 e com T posicionado no símbolo não branco mais à esquerda na fita. O primeiro passo, usando a quíntupla $(s_0, 0, s_0, 0, R)$, lê o 0 na célula não branca mais à esquerda, permanece no estado s_0, escreve 0 nesta célula e se move uma célula para a direita.

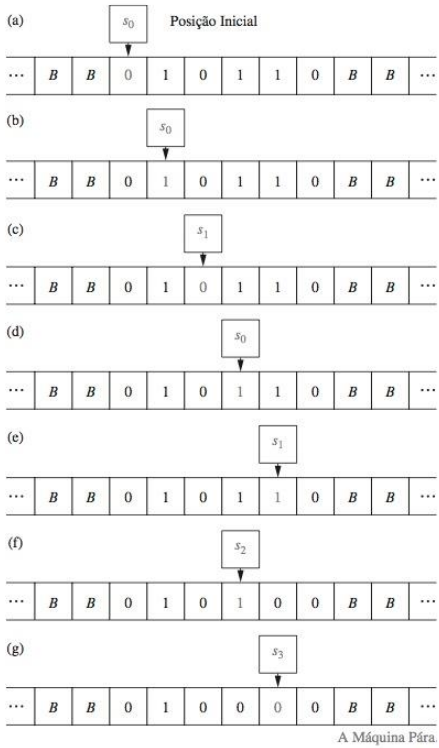


FIGURA 2 **Os Passos Produzidos pela Execução de T na Fita da Figura 1.**

O segundo passo, usando a quíntupla $(s_0, 1, s_1, 1, R)$, lê o 1 na célula corrente, entra o estado s_1, escreve um 1 nesta célula e se move uma célula para a direita. O terceiro passo, usando a quíntupla $(s_1, 0, s_0, 0, R)$, lê o 0 na célula corrente, entra o estado s_0, escreve um 0 nesta célula e se move uma célula para a direita. O quarto passo, usando a quíntupla $(s_0, 1, s_1, 1, R)$, lê o 1 na célula corrente, entra o estado s_1, escreve um 1 nesta célula e se move para a direita uma célula. O quinto passo, usando a quíntupla $(s_1, 1, s_2, 0, L)$, lê o 1 na célula corrente, entra o estado s_2, escreve um 0 nesta célula e se move uma célula para a esquerda. O sexto passo, usando a quíntupla $(s_2, 1, s_3, 0, R)$, lê o 1 na célula corrente, entra o estado s_3, escreve um 0 nesta célula e se move uma célula para a direita. Finalmente, no sétimo passo, a máquina pára porque não há nenhuma quíntupla começando com o par $(s_3, 0)$ na descrição da máquina. Os passos são mostrados na Figura 2.

Observe que T muda o primeiro par de 1s consecutivos na fita para 0s e então pára. ◀

Usando Máquinas de Turing para Reconhecer Conjuntos

As máquinas de Turing podem ser usadas para reconhecer conjuntos. Para fazer isso, é necessário definir o conceito de estado final como segue. Um *estado final* de uma máquina de Turing T é um estado que não é o primeiro estado em nenhuma quíntupla na descrição de T usando quíntuplas (por exemplo, o estado s_3 no Exemplo 1).

Podemos agora definir o que significa uma máquina de Turing reconhecer uma seqüência. Dada uma seqüência, escrevemos os símbolos consecutivos nesta seqüência em células consecutivas.

DEFINIÇÃO 2 Seja V um subconjunto do alfabeto I. Uma máquina de Turing $T = (S, I, f, s_0)$ *reconhece* uma seqüência x em V^* se e somente se T, começando na posição inicial quando x estiver escrita na fita, pára em um estado final. Dizemos que T reconhece um subconjunto A de V^* se x é reconhecida por T se e somente se x pertence a A.

Observe que para reconhecer um subconjunto A de V^* podemos usar símbolos que não estão em V. Isto significa que o alfabeto de entrada I pode incluir símbolos que não estão em V. Estes símbolos extras em geral são usados como marcadores (veja o Exemplo 3).

Quando uma máquina de Turing T *não* reconhece uma seqüência x em V^*? A resposta é que x não é reconhecida se T não parar ou parar em um estado que não é final quando ela opera em uma fita que contenha os símbolos de x em células consecutivas, começando na posição inicial. (O leitor deve entender que esta é uma das possíveis maneiras de definir como reconhecer conjuntos usando máquinas de Turing.)

Ilustramos este conceito com o Exemplo 2.

EXEMPLO 2 Encontre uma máquina de Turing que reconheça o conjunto das seqüências de bits que têm um 1 como seu segundo bit, ou seja, o conjunto regular $\mathbf{(0 \cup 1)1(0 \cup 1)^*}$.

Solução: Queremos uma máquina de Turing que, começando na célula não branca mais à esquerda na fita, se move para a direita e determina se o segundo símbolo é um 1. Se o segundo símbolo for 1, a máquina deve se mover para um estado final. Se o segundo símbolo não for 1, a máquina não deve parar ou deve parar em um estado não final.

Para construir essa máquina, incluímos as quíntuplas $(s_0, 0, s_1, 0, R)$ e $(s_0, 1, s_1, 1, R)$ para ler o primeiro símbolo e colocar a máquina de Turing no estado s_1. A seguir, incluímos as quíntuplas $(s_1, 0, s_2, 0, R)$ e $(s_1, 1, s_3, 1, R)$ para ler o segundo símbolo e se mover ou para um estado s_2 se este símbolo for um 0, ou para o estado s_3 se o símbolo for um 1. Não queremos reconhecer seqüências que tenham um 0 como seu segundo bit, de modo que s_2 não deve ser um estado final. Queremos que s_3 seja um estado final. Assim, podemos incluir a quíntupla $(s_2, 0, s_2, 0, R)$. Como não queremos reconhecer a seqüência vazia ou a seqüência com um bit, também incluímos as quíntuplas $(s_0, B, s_2, 0, R)$ e $(s_1, B, s_2, 0, R)$.

A máquina de Turing T que consiste nas sete quíntuplas listadas aqui vai acabar no estado s_3 se e somente se a seqüência de bits tiver pelo menos dois bits e o segundo bit da seqüência de entrada for 1. Se a seqüência de bits contiver menos de dois bits ou se o segundo bit não for um 1, a máquina acabará no estado não final s_2. ◀

Dado um conjunto regular, poderá ser construída uma máquina de Turing que se move sempre para a direita para reconhecer este conjunto (como no Exemplo 2). Para construir a máquina de Turing, primeiro encontre um autômato finito que reconheça o conjunto e a seguir construa uma máquina de Turing usando a função de transição da máquina de estados finitos, sempre se movendo para a direita.

Mostraremos agora como construir uma máquina de Turing que reconheça um conjunto não regular.

EXEMPLO 3 Encontre uma máquina de Turing que reconheça o conjunto $\{0^n1^n \mid n \geq 1\}$.

Solução: Para construir essa máquina, usaremos um símbolo de fita auxiliar M como um marcador. Temos $V = \{0, 1\}$ e $I = \{0, 1, M\}$. Gostaríamos de reconhecer apenas um subconjunto das seqüências em V^*. Teremos um único estado final, s_6. A máquina de Turing sucessivamente substitui um 0 na posição mais a esquerda da seqüência por um M e um 1 na posição mais à direita da seqüência por um M, varrendo para frente e para trás, terminando em um estado final se e somente se a seqüência consistir em um bloco de 0s seguido por um bloco com o mesmo número de 1s.

Embora isso seja fácil de descrever e fácil de executar por uma máquina de Turing, a máquina que precisamos usar é um tanto complicada. Usamos o marcador M para seguir os símbolos mais à esquerda e mais à direita que já tivermos examinado. As quíntuplas que usamos são $(s_0, 0, s_1, M, R)$, $(s_1, 0, s_1, 0, R)$, $(s_1, 1, s_1, 1, R)$, (s_1, M, s_2, M, L), (s_1, B, s_2, B, L), $(s_2, 1, s_3, M, L)$, $(s_3, 1, s_3, 1, L)$, $(s_3, 0, s_4, 0, L)$, (s_3, M, s_5, M, R), $(s_4, 0, s_4, 0, L)$, (s_4, M, s_0, M, R) e (s_5, M, s_6, M, R). Por exemplo, a seqüência 000111 se tornaria sucessivamente $M00111$, $M0011M$, $MM011M$, $MM01MM$, $MMM1MM$, $MMMMMM$ à medida que a máquina opera até que ela pare. Apenas as mudanças foram mostradas, uma vez que a maioria dos passos deixa a seqüência inalterada.

Deixamos para o leitor (Exercício 13, no final desta seção) a explicação das ações dessa máquina de Turing e a explicação de por que ela reconhece o conjunto $\{0^n1^n \mid n \geq 1\}$. ◀

Pode ser mostrado que um conjunto é reconhecido por uma máquina de Turing se e somente se ele puder ser gerado por uma gramática de tipo 0, ou, em outras palavras, se o conjunto for gerado por uma gramática de estrutura de frase. A demonstração não será apresentada aqui.

Calculando Funções com Máquinas de Turing

Uma máquina de Turing pode ser pensada como um computador que encontra os valores de uma função parcial. Para ver isso, suponha que a máquina de Turing T, quando for dada a seqüência x como entrada, pára com a seqüência y em sua fita. Podemos então definir $T(x) = y$. O domínio de T é o conjunto das seqüências para as quais T pára; $T(x)$ não está definida se T não parar quando x for dada como entrada. Pensar em uma máquina de Turing como uma máquina que calcula os valores de uma função em seqüências é útil, mas como podemos usar máquinas de Turing para calcular funções definidas nos inteiros, ou em pares de inteiros, ou em triplas de inteiros, e assim por diante?

Para considerar uma máquina de Turing como um computador que calcule funções do conjunto das k-uplas de inteiros não negativos para o conjunto dos inteiros não negativos (essas funções são chamadas de **funções number-theoretic**), precisamos de uma forma de representar k-uplas de inteiros em uma fita. Para fazer isso, usamos **representações unárias** dos inteiros. Representamos o inteiro não negativo n por uma seqüência de $n + 1$ 1s de modo que, por exemplo, 0 é representado pela seqüência 1 e 5 é representado pela seqüência 111111. Para representar a k-upla $(n_1, n_2, \ldots, n_k)$, usamos uma seqüência de $n_1 + 1$ 1s, seguida por um asterisco, seguido por uma seqüência de $n_2 + 1$ 1s, seguida por um asterisco, e assim por diante, terminando com uma seqüência de $n_k + 1$ 1s. Por exemplo, para representar a quádrupla $(2, 0, 1, 3)$ usamos a seqüência $111 * 1 * 11 * 1111$.

Podemos considerar uma máquina de Turing T como computando uma seqüência de funções *number-theoretic* $T, T^2, \ldots, T^k, \ldots$. A função T^k é definida pela ação de T nas k-uplas de inteiros mostrados pela representação unária de inteiros separada por asteriscos.

EXEMPLO 4 Construa uma máquina de Turing para somar dois inteiros não negativos.

Solução: Precisamos construir uma máquina de Turing T que calcule a função $f(n^1, n^2) = n_1 + n_2$. O par (n_1, n_2) é representado por uma seqüência de $n_1 + 1$ 1s seguida por um asterisco seguido por $n_2 + 1$ 1s. A máquina T deveria tomar isso como entrada e produzir como saída uma fita com $n_1 + n_2 + 1$ 1s. Uma forma de fazer isso é como segue. A máquina começa no 1 mais à esquerda da seqüência de entrada e executa passos para apagar este 1, parando se $n_1 = 0$, de modo que não há mais nenhum 1 antes do asterisco, substitui o asterisco pelo 1 remanescente mais à esquerda e então pára. Podemos usar estas quíntuplas para fazer isso: $(s_0, 1, s_1, B, R)$, $(s_1, *, s_3, B, R)$, $(s_1, 1, s_2, B, R)$, $(s_2, 1, s_2, 1, R)$ e $(s_2, *, s_3, 1, R)$. ◀

Infelizmente, a construção de máquinas de Turing para calcular funções relativamente simples pode ser extremamente trabalhosa. Por exemplo, uma máquina de Turing para multiplicar dois inteiros não negativos encontrada em muitos livros tem 31 quíntuplas e 11 estados. Se é desafiador construir máquinas de Turing para calcular mesmo funções relativamente simples, que esperança temos de construir máquinas de Turing para funções mais complicadas? Uma maneira de simplificar este problema é usar uma máquina de Turing multifita que usa mais de uma fita simultaneamente e construir essas máquinas para a composição de funções. Pode ser mostrado que para qualquer máquina de Turing multifita existe uma máquina de Turing com uma única fita que pode fazer a mesma coisa.

Tipos Diferentes de Máquinas de Turing

Existem muitas variações na definição de uma máquina de Turing. Podemos expandir os recursos de uma máquina de Turing em uma grande variedade de maneiras. Por exemplo, podemos permitir que uma máquina de Turing se mova para a direita, para a esquerda, ou que não se mova em cada passo. Podemos permitir que uma máquina de Turing opere em fitas múltiplas, usando $(2 + 3n)$-uplas para descrever a máquina de Turing quando são usadas n fitas. Podemos permitir que a fita seja bidimensional, em que a cada passo nos movemos para cima, para baixo, para a direita ou para a esquerda, e não apenas para a direita ou para a esquerda como fazemos na fita unidimensional. Podemos permitir cabeçotes de fita múltiplos que lêem células diferentes simultaneamente. Além disso, podemos permitir que uma máquina de Turing seja **não determinística**, permitindo que um par (estado, símbolo na fita) apareça possivelmente como primeiro elemento em mais de uma quíntupla da máquina de Turing. Podemos também reduzir os recursos de uma máquina de Turing de maneiras diferentes. Por exemplo, podemos restringir a fita a ser infinita em apenas uma dimensão ou podemos restringir o alfabeto da fita para ter apenas dois símbolos. Todas as variações das máquinas de Turing foram estudadas em detalhe.

O ponto crucial é que não importa qual dessas variações usemos, ou mesmo quais combinações de variações usemos, nunca aumentamos ou diminuímos o poder dessas máquinas. Qualquer coisa que uma dessas variações pode fazer pode ser feita pela máquina de Turing definida nesta seção, e vice-versa. A razão pela qual estas variações são úteis é que às vezes elas tornam a execução de alguma tarefa particular muito mais simples do que se fosse usada a máquina de Turing descrita na Definição 1. Elas nunca aumentam os recursos da máquina. Às vezes é útil ter uma grande variedade de máquinas de Turing para trabalhar. Por exemplo, uma forma de mostrar que para toda máquina de Turing não determinística existe uma máquina de Turing determinística que reconhece a mesma linguagem é usar uma máquina determinística com três fitas. (Para detalhes sobre as variações das máquinas de Turing e demonstrações de suas equivalências, veja [HoMoU101].)

Além de introduzir a noção de máquina de Turing, Turing também mostrou que é possível construir uma única máquina de Turing que pode simular os cálculos de qualquer uma dessas máquinas quando for dada uma codificação desta máquina alvo e sua entrada. Essa máquina é chamada de **máquina de Turing universal**. (Veja um livro sobre a teoria da computação, tal como [Si06], para mais informações sobre as máquinas de Turing universais.)

A Tese de Church–Turing

Links

As máquinas de Turing são relativamente simples. Elas podem ter apenas um número finito de estados e podem ler e escrever apenas um símbolo de cada vez em uma fita unidimensional. Contudo, ocorre que as máquinas de Turing são extremamente poderosas. Vimos que podem ser construídas para somar números e para multiplicar números. Embora possa ser difícil realmente construir uma máquina de Turing para calcular uma função particular que possa ser calculada por um algoritmo, essa máquina sempre pode ser encontrada. Este era o objetivo original de Turing quando inventou suas máquinas.

Além disso, existe uma tremenda quantidade de evidência para a **tese de Church–Turing**, que afirma que dado qualquer problema que possa ser resolvido com um algoritmo efetivo, existe uma máquina de Turing que pode resolver este problema. A razão de isso ser chamado de *tese* e não de teorema é que o conceito de solvabilidade por um algoritmo efetivo é informal e impreciso, em oposição à noção de solvabilidade por uma máquina de Turing, que é formal e precisa. De qualquer forma, certamente qualquer problema que possa ser resolvido usando um computador com um programa escrito em qualquer linguagem, talvez usando uma quantidade ilimitada de memória, deveria ser considerado efetivamente solúvel. (Observe que as máquinas de Turing têm memória ilimitada, diferentemente dos computadores do mundo real, os quais têm apenas uma quantidade finita de memória.)

Muitas teorias formais diferentes foram desenvolvidas para capturar a noção de computabilidade efetiva. Estas incluem a teoria de Turing e o cálculo lambda de Church, bem como as teorias propostas por Stephen Kleene e por E. L. Post. Essas teorias parecem superficialmente bastantes diferentes. O fato surpreendente é que pode ser mostrado que elas são equivalentes, através da demonstração de que elas definem exatamente a mesma classe de funções. Com esta evidência, parece que as idéias originais de Turing, formuladas antes da invenção dos computadores modernos, descrevem os recursos finais destas máquinas. O leitor interessado deveria consultar livros sobre a teoria da computação, tais como [HoMoU101] e [Si96], para uma discussão destas teorias diferentes e sua equivalência.

No resto desta seção, vamos explorar brevemente algumas das conseqüências da tese de Church–Turing e descrever a importância das máquinas de Turing no estudo da complexidade de algoritmos. Nosso objetivo será introduzir algumas idéias importantes da ciência da computação para atrair o estudante interessado para mais estudos. Cobriremos rapidamente um extenso material sem fornecer detalhes explícitos. Nossa discussão também ligará alguns dos conceitos discutidos em partes anteriores deste livro à teoria da computação.

Complexidade Computacional, Computabilidade e Decidibilidade

Por todo este livro discutimos a complexidade computacional de uma grande variedade de problemas. Descrevemos a complexidade desses problemas em termos do número de operações usadas pelos algoritmos mais eficientes para resolvê-los. As operações básicas usadas pelos algoritmos diferem consideravelmente; medimos a complexidade de algoritmos diferentes em termos de operações com bits, comparações de inteiros, operações aritméticas, e assim por diante. Na Seção 3.3, definimos várias classes de problemas em termos de sua complexidade computacional. Entretanto, estas definições não eram precisas, pois os tipos de operações usadas para medir sua complexidade variavam muito drasticamente. As máquinas de Turing fornecem uma maneira de tornar o conceito de complexidade computacional preciso. Pela tese de Church–Turing, sabemos que se um problema puder ser resolvido usando um algoritmo efetivo, então existe uma máquina de Turing que pode resolver esse problema. Quando uma máquina de Turing é usada para resolver um problema, a entrada do problema é codificada como uma seqüência de símbolos que é escrita na fita desta máquina de Turing. Como codificamos a entrada depende do domínio desta entrada. Por exemplo, como já vimos, podemos codificar um inteiro positivo usando uma seqüência de 1s. Podemos também descobrir maneiras de expressar pares de inteiros, inteiros negativos, e assim por diante. Analogamente, para algoritmos em grafos, precisamos de uma maneira de codificar grafos como seqüências de símbolos. Isso pode ser feito de muitas maneiras e pode basear-se em listas de adjacência ou em matrizes de adjacência. (Omitiremos os detalhes de como isso é feito.) Entretanto, a maneira de a entrada ser codificada não importa, contanto que ela seja relativamente eficiente, uma vez que uma máquina de Turing pode sempre mudar uma codificação para uma

outra codificação. Agora, usaremos este modelo para tornar precisas algumas das noções relativas à complexidade computacional que foram introduzidas informalmente na Seção 3.3.

O tipo de problemas que são estudados mais facilmente usando as máquinas de Turing são aqueles que podem ser respondidos por um "sim" ou por um "não".

DEFINIÇÃO 3 Um *problema de decisão* pergunta se afirmações de uma classe particular de afirmações são verdadeiras. Os problemas de decisão também são conhecidos como *problemas sim-ou-não*.

Dado um problema de decisão, gostaríamos de saber se existe um algoritmo que pode determinar se afirmações da classe de afirmações a que ele se refere são verdadeiras. Por exemplo, considere a classe de afirmações, cada uma das quais pergunta se um inteiro particular n é primo. Este é um problema de decisão, porque a resposta a esta questão "n é primo?" é sim ou não. Dado este problema de decisão, podemos perguntar se existe um algoritmo que possa decidir se cada uma das afirmações no problema de decisão é verdadeira, ou seja, dado um inteiro n, a decisão de se n é primo. A resposta é que existe esse algoritmo. Em particular, na Seção 3.5 discutimos o algoritmo que determina se um inteiro positivo n é primo, verificando se ele é divisível pelos primos que não excedam sua raiz quadrada. (Existem muitos outros algoritmos para determinar se um inteiro positivo é primo.) O conjunto de entradas, para as quais a resposta do problema sim–não é "sim", é um subconjunto do conjunto de entradas possíveis, ou seja, é um subconjunto do conjunto de seqüências do alfabeto de entrada. Em outras palavras, resolver um problema sim-não é o mesmo que reconhecer a linguagem que consiste em todas as seqüências de bits que representam valores de entrada do problema que levam à resposta "sim". Conseqüentemente, resolver um problema sim–não é o mesmo que reconhecer a linguagem correspondente aos valores de entrada para os quais a resposta do problema é "sim".

DECIDIBILIDADE Quando existe um algoritmo efetivo que decide se casos de um problema de decisão são verdadeiros, dizemos que este problema é **solúvel** ou **decidível**. Por exemplo, o problema de determinar se um inteiro positivo é primo é um problema solúvel. Entretanto, se não existir nenhum algoritmo efetivo para resolver o problema, dizemos que o problema é **insolúvel** ou **indecidível**. Para mostrar que um problema de decisão é solúvel, precisamos apenas construir um algoritmo que possa determinar se afirmações de uma classe particular são verdadeiras. Por outro lado, para mostrar que um problema de decisão é insolúvel, precisamos demonstrar que não existe esse algoritmo. (O fato de termos tentado encontrar esse algoritmo, mas falhado, não demonstra que esse problema é insolúvel.)

Pode parecer que estudando apenas problemas de decisão, estamos estudando apenas um pequeno conjunto dos problemas de interesse. Entretanto, a maioria dos problemas pode ser reenunciada como um problema de decisão. Reenunciar os tipos de problemas que estudamos neste livro, como problemas de decisão, pode ser bastante complicado, de modo que não entraremos em detalhes sobre esse processo aqui. O leitor interessado pode consultar referências sobre a teoria de computação, tal como [Wo87], a qual, por exemplo, explica como reenunciar o problema do caixeiro viajante (descrito na Seção 9.6) como um problema de decisão. (Para reenunciar o problema do caixeiro viajante como um problema de decisão, consideramos primeiro o problema de decisão que pergunta se existe um ciclo hamiltoniano de peso que não exceda k, em que k é um inteiro positivo. Com algum esforço adicional é possível usar respostas a esta questão para diferentes valores de k para encontrar o menor peso possível de um ciclo hamiltoniano.)

Na Seção 3.1, introduzimos um problema de parada e demonstramos que é um problema insolúvel. A discussão foi um tanto informal, pois a noção de procedimento não estava precisamente definida. Uma definição precisa de um problema de parada pode ser feita em termos de máquinas de Turing.

DEFINIÇÃO 4 O *problema de parada* é o problema de decisão que pergunta se uma máquina de Turing T eventualmente pára quando for dada uma seqüência de entrada x.

Com esta definição de problema de parada, temos o Teorema 1.

TEOREMA 1 O problema de parada é um problema de decisão insolúvel. Ou seja, não existe nenhuma máquina de Turing que, quando dada uma codificação de uma máquina de Turing T e sua seqüência de entrada x como entrada, pode determinar se T eventualmente pára quando for iniciada com x escrita em sua fita.

A demonstração do Teorema 1 dada na Seção 3.1 para a definição informal do problema de parada ainda se aplica aqui.

Outros exemplos de problemas insolúveis incluem:

(a) o problema de determinar se duas gramáticas livres de contexto geram o mesmo conjunto de seqüências;
(b) o problema de determinar se um dado conjunto de tijolos pode ser usado com repetição permitida para cobrir o plano todo sem superposição; e
(c) o *Décimo Problema de Hilbert*, que pergunta se existem soluções inteiras de uma dada equação polinomial com coeficientes inteiros. (Esta pergunta ocorre em décimo lugar na famosa lista de 23 problemas que Hilbert propôs em 1900. Hilbert previu que o trabalho é realizado para resolver estes problemas ajudaria a desenvolver mais o progresso da matemática no século XX. O fato de o Décimo Problema de Hilbert ser insolúvel foi estabelecido em 1970, por Yuri Matiyasevich.)

COMPUTABILIDADE Uma função que pode ser calculada por uma máquina de Turing é chamada de **computável** e uma função que não pode ser calculada por uma máquina de Turing é chamada de **não computável**. É bastante simples, usando um argumento de contabilidade, mostrar que existem funções *number-theoretic* que não são computáveis (veja o Exercício 47 da Seção 2.4). Entretanto, não é tão fácil produzir de fato essa função. A **função busy beaver** definida no preâmbulo do Exercício 31 no final desta seção é um exemplo de uma função não computável. Uma maneira de mostrar que a função *busy beaver* não é computável é mostrar que ela cresce mais rapidamente do que qualquer função computável. (Veja o Exercício 32.)

Observe que todo problema de decisão pode ser reformulado como o problema de calcular uma função, a saber, a função que tem o valor 1 quando a resposta do problema é "sim" e que tem o valor 0 quando a resposta do problema é "não". Um problema de decisão é solúvel se e somente se a função correspondente construída desta maneira é computável.

AS CLASSES P E NP Na Seção 3.3, definimos informalmente as classes de problemas chamadas de P e NP. Agora, somos capazes de definir estes conceitos precisamente, usando as noções de máquinas de Turing determinísticas e não determinísticas.

Primeiro elaboramos a diferença entre uma máquina de Turing determinística e uma máquina de Turing não determinística. As máquinas que estudamos nesta seção foram todas determinísticas. Em uma máquina de Turing determinística $T = (S, I, f, s_0)$, as regras de transição são definidas pelas funções parciais f de $S \times I$ para $S \times I \times \{R, L\}$. Conseqüentemente, quando as regras de transição das máquinas são representadas como quíntuplas da forma (s, x, s', x', d), em que s é o estado corrente, x é o símbolo corrente na fita, s' é o estado seguinte, x' é o símbolo que substitui x na fita e d é a direção em que a máquina se move na fita, nenhum par de regras de transição começa com o mesmo par (s, x).

Em uma máquina de Turing não determinística, os passos permitidos são definidos usando uma relação que consiste em quíntuplas em vez de usar uma função parcial. A restrição de que nenhum par de regras de transição comece com o mesmo par (s, x) é eliminada; ou seja, pode existir mais de uma regra de transição que comece com cada par (estado, símbolo na fita). Conseqüentemente, em uma máquina de Turing não determinística, existe uma escolha de transições para alguns pares de estado corrente e símbolo sendo lido na fita. Em cada passo da operação de uma máquina de Turing não determinística, a máquina toma uma das diferentes escolhas de regras de transição que começam com o par corrente estado e símbolo na fita. Essa escolha pode ser considerada como uma "conjectura" de qual passo usar. Do mesmo modo que para as máquinas de Turing determinísticas, uma máquina de Turing não determinística pára quando não existir

nenhuma regra de transição na sua definição que comece com o estado corrente e o símbolo na fita. Dada uma máquina de Turing não determinística T, dizemos que uma seqüência x é reconhecida por T se e somente se existir alguma seqüência de transições de T que acabe em um estado final quando a máquina começar na posição inicial com x escrito na fita. A máquina de Turing não determinística T reconhece o conjunto A se x for reconhecida por T se e somente se $x \in A$. Dizemos que a máquina de Turing não determinística T resolve um problema de decisão se ela reconhece o conjunto que consiste de todos os valores de entrada para os quais a resposta ao problema de decisão é sim.

DEFINIÇÃO 5 Um problema de decisão está em P, *a classe dos problemas polinomiais no tempo,* se ele puder ser resolvido por uma máquina de Turing determinística em tempo polinomial em termos do tamanho da sua entrada. Ou seja, um problema de decisão está em P se existir uma máquina de Turing determinística T que resolva o problema de decisão e um polinômio $p(n)$ tal que para todos os inteiros n, T pára em um estado final depois de não mais que $p(n)$ transições sempre que a entrada de T for uma seqüência de comprimento n. Um problema de decisão está em ***NP, a classe dos problemas polinomiais no tempo não determinísticos,*** se ele puder ser resolvido por uma máquina de Turing não determinística em tempo polinomial em termos do tamanho de sua entrada. Ou seja, um problema de decisão está em NP se existir uma máquina de Turing não determinística T que resolva o problema e um polinômio $p(n)$ tal que para todos os inteiros n, T pára para todas as escolhas de transições depois de não mais que $p(n)$ transições, sempre que a entrada de T for uma seqüência de comprimento n.

Os problemas em P são ditos **tratáveis**, enquanto os problemas que não estão em P são ditos **intratáveis**. Para que um problema esteja em P, deve existir uma máquina de Turing determinística que pode decidir em tempo polinomial se uma afirmação particular da classe a que se refere o problema de decisão é verdadeira. Por exemplo, determinar se um item está em uma lista de n elementos é um problema tratável. (Não forneceremos detalhes de como este fato pode ser mostrado; as idéias básicas usadas nas análises dos algoritmos anteriormente no texto podem ser adaptadas quando as máquinas de Turing são empregadas.) Surpreendentemente, o problema de determinar se um inteiro positivo n é primo também é tratável (um fato que foi demonstrado apenas em 2002 por Manindra Agrawal, Neeraj Kayal e Nitin Saxena [AgKaSa02]). Para um problema estar em NP, é necessário apenas que exista uma máquina de Turing não determinística que, quando for dada uma afirmação verdadeira do conjunto de afirmações a que se refere o problema, pode verificar sua veracidade em tempo polinomial, fazendo uma conjectura correta em cada passo do conjunto dos passos permitidos correspondentes ao estado e símbolo na fita correntes. O problema de determinar se um dado grafo tem um ciclo hamiltoniano é um problema NP, pois uma máquina de Turing não determinística pode facilmente verificar que um ciclo simples em um grafo passa pelos vértices exatamente uma vez. Ela pode fazer isto realizando uma série de conjecturas corretas correspondendo à adição sucessiva de arestas para formar o ciclo. Como toda máquina de Turing determinística pode ser considerada como uma máquina de Turing não determinística, na qual cada par (estado, símbolo na fita) ocorre em exatamente uma regra de transição definindo a máquina, todo problema em P também está em NP. Em símbolos, $P \subseteq NP$.

Uma das questões abertas mais intrincadas na ciência da computação teórica é se todo problema em NP também está em P, ou seja, se $P = NP$. Este problema é um dos sete problemas

Links

ALONZO CHURCH (1903–1995) Alonzo Church nasceu em Washington, D.C. Ele estudou em Göttingen, sob a orientação de Hilbert, e mais tarde em Amsterdã. Foi membro do corpo docente da Universidade de Princeton de 1927 a 1967, quando se mudou para a UCLA. Church foi um dos membros fundadores da Association for Symbolic Logic. Ele deu várias contribuições substanciais para a teoria da computabilidade, incluindo a sua solução para o problema de decisão, sua invenção do cálculo lambda e, é claro, seu enunciado do que agora é conhecido como tese de Church–Turing. Entre os alunos de Church estiveram Stephen Kleene e Alan Turing. Ele publicou artigos até depois do seu nonagésimo aniversário.

conhecidos como *Problemas do Milênio*, os quais devem desempenhar um papel similar para a matemática do século XXI ao que os problemas de Hilbert desempenharam para a matemática do século XX. (Veja [De02] para mais informações sobre os Problemas do Milênio.) Como mencionado na Seção 3.3, existe uma importante classe de problemas, a classe dos problemas NP completos, tal que um problema está nesta classe se ele está na classe NP e se for possível mostrar que ele também está na classe P, então *todo* problema na classe NP também deve estar na classe P. Ou seja, um problema é NP completo se a existência de um algoritmo polinomial no tempo para resolvê-lo implica a existência de um algoritmo polinomial no tempo para resolver qualquer problema em NP. Neste livro, discutimos diversos problemas NP completos diferentes, tal como determinar se um grafo simples tem um ciclo hamiltoniano e determinar se uma proposição em n variáveis é uma tautologia.

Exercícios

1. Seja T uma máquina de Turing definida pelas quíntuplas $(s_0, 0, s_1, 1, R)$, $(s_0, 1, s_1, 0, R)$, $(s_0, B, s_1, 0, R)$, $(s_1, 0, s_2, 1, L)$, $(s_1, 1, s_1, 0, R)$ e $(s_1, B, s_2, 0, L)$. Para cada uma destas fitas iniciais, determine a fita final quando T parar, supondo que T comece na posição inicial.

a)

...	B	B	0	0	1	1	B	B	...

b)

...	B	B	1	0	1	B	B	B	...

c)

...	B	B	1	1	B	0	1	B	...

d)

...	B	B	B	B	B	B	B	B	...

2. Seja T uma máquina de Turing definida pelas quíntuplas $(s_0, 0, s_1, 0, R)$, $(s_0, 1, s_1, 0, L)$, $(s_0, B, s_1, 1, R)$, $(s_1, 0, s_2, 1, R)$, $(s_1, 1, s_1, 1, R)$, $(s_1, B, s_2, 0, R)$ e $(s_2, B, s_3, 0, R)$. Para cada uma destas fitas iniciais, determine a fita final quando T parar, supondo que T comece na posição inicial.

a)

...	B	B	0	1	0	1	B	B	...

b)

...	B	B	1	1	1	B	B	B	...

c)

...	B	B	0	0	B	0	0	B	...

d)

...	B	B	B	B	B	B	B	B	...

3. O que a máquina de Turing descrita pelas quíntuplas $(s_0, 0, s_0, 0, R)$, $(s_0, 1, s_1, 0, R)$, (s_0, B, s_2, B, R), $(s_1, 0, s_1, 0, R)$, $(s_1, 1, s_0, 1, R)$ e (s_1, B, s_2, B, R) faz quando é dado

a) 11 como entrada?

b) uma seqüência de bits arbitrária como entrada?

4. O que a máquina de Turing descrita pelas quíntuplas $(s_0, 0, s_0, 1, R)$, $(s_0, 1, s_0, 1, R)$, (s_0, B, s_1, B, L), $(s_1, 1, s_2, 1, R)$ faz quando é dado

a) 101 como entrada?

b) uma seqüência de bits arbitrária como entrada?

5. O que a máquina de Turing descrita pelas quíntuplas $(s_0, 1, s_1, 0, R)$, $(s_1, 1, s_1, 1, R)$, $(s_1, 0, s_2, 0, R)$, $(s_2, 0, s_3, 1, L)$, $(s_2, 1, s_2, 1, R)$, $(s_3, 1, s_3, 1, L)$, $(s_3, 0, s_4, 0, L)$, $(s_4, 1, s_4, 1, L)$ e $(s_4, 0, s_0, 1, R)$ faz quando é dado

a) 11 como entrada?

b) uma seqüência de bits que consiste inteiramente de 1s como entrada?

6. Construa uma máquina de Turing com símbolos de fita 0, 1 e B que, quando dada uma seqüência de bits como entrada, adiciona um 1 ao final da seqüência de bits e não muda nenhum dos outros símbolos na fita.

7. Construa uma máquina de Turing com símbolos de fita 0, 1 e B que, quando dada uma seqüência de bits como entrada, substitui o primeiro 0 por um 1 e não muda nenhum dos outros símbolos na fita.

8. Construa uma máquina de Turing com símbolos de fita 0, 1 e B que, quando dada uma seqüência de bits como entrada, substitui todos os 0s por 1s e não muda nenhum dos outros 1s na fita.

9. Construa uma máquina de Turing com símbolos de fita 0, 1 e B que, quando dada uma seqüência de bits como entrada, substitui todos, exceto o 1 mais à esquerda, por 0s e não muda nenhum dos outros símbolos na fita.

10. Construa uma máquina de Turing com símbolos de fita 0, 1 e B que, quando dada uma seqüência de bits como entrada, substitui os dois primeiros 1s consecutivos por 0 e não muda nenhum dos outros símbolos na fita.

11. Construa uma máquina de Turing que reconheça o conjunto de todas as seqüências de bits que terminam com um 0.

12. Construa uma máquina de Turing que reconheça o conjunto de todas as seqüências de bits que contenham pelo menos dois 1s.

13. Construa uma máquina de Turing que reconheça o conjunto de todas as seqüências de bits que contenham um número par de 1s.

14. Mostre em cada passo o conteúdo da fita na máquina de Turing do Exemplo 3 começando com cada uma destas seqüências.

a) 0011 **b)** 00011 **c)** 101100 **d)** 000111

15. Explique por que a máquina de Turing do Exemplo 3 reconhece uma seqüência de bits se e somente se esta seqüência for da forma 0^n1^n para algum inteiro positivo n.

*16. Construa uma máquina de Turing que reconheça o conjunto $\{0^{2n}1^n \mid n \geq 0\}$.

*17. Construa uma máquina de Turing que reconheça o conjunto $\{0^n1^n2^n \mid n \geq 0\}$.

18. Construa uma máquina de Turing que calcule a função $f(n) = n + 2$ para todos os inteiros não negativos n.

19. Construa uma máquina de Turing que calcule a função $f(n) = n - 3$ se $n \geq 3$ e $f(n) = 0$ para $n = 0, 1, 2$ para todos os inteiros não negativos n.

20. Construa uma máquina de Turing que calcule a função $f(n) = n$ **mod** 3.

21. Construa uma máquina de Turing que calcule a função $f(n) = 3$ se $n \geq 5$ e $f(n) = 0$ se $n = 0, 1, 2, 3$ ou 4.

22. Construa uma máquina de Turing que calcule a função $f(n) = 2n$ para todos os inteiros não negativos n.

23. Construa uma máquina de Turing que calcule a função $f(n) = 3n$ para todos os inteiros não negativos n.

24. Construa uma máquina de Turing que calcule a função $f(n_1, n_2) = n_2 + 2$ para todos os pares de inteiros não negativos n_1 e n_2.

*25. Construa uma máquina de Turing que calcule a função $f(n_1, n_2) = \min(n_1, n_2)$ para todos os inteiros não negativos n_1 e n_2.

26. Construa uma máquina de Turing que calcule a função $f(n_1, n_2) = n_1 + n_2 + 1$ para todos os inteiros não negativos n_1 e n_2.

Suponha que T_1 e T_2 sejam máquinas de Turing com conjuntos disjuntos de estados S_1 e S_2 e com funções de transição f_1 e f_2, respectivamente. Podemos definir a máquina de Turing T_1T_2, a **composta** de T_1 e T_2, como segue. O conjunto dos estados de T_1T_2 é $S_1 \cup S_2$. T_1T_2 começa no estado inicial de S_1. Ela primeiro executa as transições de T_1 usando f_1 até, mas não incluindo, o passo no qual T_1 pararia. Então, para todos os movimentos para os quais T_1 pára, ela executa as mesmas transições de T_1, exceto que ela se move para o estado inicial de T_2. Daí para diante, os movimentos de T_1T_2 são os mesmos que os movimentos de T_2.

27. Construa uma máquina de Turing que calcule a função $f(n) = 2n + 2$, encontrando a composta das máquinas de Turing que você construiu nos exercícios 18 e 22.

28. Construa uma máquina de Turing que calcule a função $f(n) = 3(n + 2) = 3n + 6$, encontrando a composta das máquinas de Turing que você construiu nos exercícios 18 e 23.

29. Quais dos seguintes problemas são problemas de decisão?
   a) Qual é o menor primo maior do que n?
   b) Um grafo G é bipartido?
   c) Dado um conjunto de seqüências, existe um autômato finito que reconheça este conjunto de seqüências?
   d) Dado um tabuleiro de xadrez e um tipo particular de poliominó (veja a Seção 1.7), pode o tabuleiro ser ladrilhado usando polominós deste tipo?

30. Quais dos seguintes problemas são problemas de decisão?
   a) A seqüência $a_1, a_2, \ldots, a_n$ de inteiros positivos está em ordem crescente?
   b) Os vértices de um grafo simples G podem ser coloridos usando três cores, de modo que nenhum par de vértices adjacentes tenha a mesma cor?
   c) Qual o vértice de maior grau em um grafo G?
   d) Dadas duas máquinas de estados finitos, estas máquinas reconhecem a mesma linguagem?

Seja $B(n)$ o número máximo de 1s que uma máquina de Turing com n estados com o alfabeto $\{1, B\}$ pode imprimir em uma fita que esteja inicialmente em branco. O problema de determinar $B(n)$ para valores particulares de n é conhecido como o **problema busy beaver.** Este problema foi inicialmente estudado por Tibor Rado, em 1962. Atualmente é conhecido que $B(2) = 4$, $B(3) = 6$ e $B(4) = 13$, mas $B(n)$ não é conhecido para $n \geq 5$. $B(n)$ cresce rapidamente; é conhecido que $B(5) \geq 4098$ e $B(6) \geq 1{,}29 \times 10^{865}$.

*31. Mostre que $B(2)$ é pelo menos 4, encontrando uma máquina de Turing com dois estados e alfabeto $\{1, B\}$ que pare com quatro 1s consecutivos na fita.

**32. Mostre que a função $B(n)$ não pode ser calculada por nenhuma máquina de Turing. [*Dica*: Suponha que exista uma máquina de Turing que calcule $B(n)$ em binário. Construa uma máquina de Turing T que, começando com uma fita em branco, escreve n em binário, calcula $B(n)$ em binário e converte $B(n)$ de binário para unário. Mostre que, para n suficientemente grande, o número de estados de T é menor do que $B(n)$, levando a uma contradição.]

Termos-chave e Resultados

TERMOS

alfabeto (ou vocabulário): um conjunto que contém elementos usados para formar seqüências

linguagem: um subconjunto do conjunto de todas as seqüências em um alfabeto

gramática de estrutura de frase (V, T, S, P): uma descrição de uma linguagem que contém um alfabeto V, um conjunto de símbolos terminais T, um símbolo inicial S e um conjunto de regras de produção P

a regra de produção $w \rightarrow w_1$: w pode ser substituído por w_1 sempre que ocorrer em uma seqüência na linguagem

$w_1 \Rightarrow w_2$ (w_2 é diretamente derivável de w_1): w_2 pode ser obtido de w_1, usando uma regra de produção para substituir uma seqüência em w_1 por uma outra seqüência

$w_1 \overset{*}{\Rightarrow} w_2$ (w_2 é derivável de w_1): w_2 pode ser obtido a partir de w_1, usando uma seqüência de regras de produção para substituir seqüências por outras seqüências

gramática de tipo 0: qualquer gramática de estrutura de frase

gramática de tipo 1: uma gramática de estrutura de frase na qual toda regra de produção é da forma $w_1 \rightarrow w_2$, em que $w_1 = lAr$ e $w_2 = lwr$, em que $A \in N$, $l, r, w \in (N \cup T)$ e $w = \lambda$, ou $w_1 = S$ e $w_2 = \lambda$ enquanto S não estiver no lado direito de outra regra de produção

gramática de tipo 2, ou livre de contexto: uma gramática de estrutura de frase na qual toda regra de produção é da forma $A \rightarrow w_1$, em que A é um símbolo não terminal

gramática de tipo 3, ou regular: uma gramática de estrutura de frase em que toda regra de produção é da forma $A \rightarrow aB$, $A \rightarrow a$ ou $S \rightarrow \lambda$, em que A e B são símbolos não terminais, S é o símbolo inicial e a é um símbolo terminal

árvore de derivação (ou parse tree): uma árvore enraizada ordenada em que a raiz representa o símbolo inicial de uma gramática de tipo 2, os vértices internos representam não terminais, as folhas representam terminais e os filhos de um vértice são os símbolos no lado direito de uma regra de produção, ordenados da esquerda para a direita, em que o símbolo representado pelo pai está no lado esquerdo

forma de Backus–Naur: uma descrição de uma gramática livre de contexto na qual todas as regras de produção que têm o mesmo não terminal como seu lado esquerdo são combinadas com os diferentes lados direitos destas regras de produção, cada uma separada por uma barra, com os símbolos não terminais englobados por colchetes angulares e o símbolo $\rightarrow$ substituído por ::=

máquina de estados finitos (S, I, O, f, g, s_0) (ou uma máquina de Mealy): uma sêxtupla que contém um conjunto S de estados, um alfabeto de entrada I, um alfabeto de saída O, uma função de transição f que associa um estado seguinte a todo par de um estado e uma entrada, uma função de saída g que associa uma saída a todo par de um estado e uma entrada, e um estado inicial s_0

***AB* (concatenação de *A* e *B*):** o conjunto de todas as seqüências formadas pela concatenação de uma seqüência em A e uma seqüência em B nesta ordem

***A** (fecho de Kleene de *A*):** o conjunto de todas as seqüências formadas por concatenações de um número arbitrário de seqüências de A

autômato finito determinístico (S, I, f, s_0, F): uma quíntupla que contém um conjunto S de estados, um alfabeto de entrada I, uma função de transição f que associa um estado seguinte a cada par de um estado e uma entrada, um estado inicial s_0 e um conjunto de estados finais F

autômato finito não determinístico (S, I, f, s_0, F): uma quíntupla que contém um conjunto S de estados, um alfabeto de entrada I, uma função de transição f que associa um conjunto de possíveis estados seguintes a cada par de um estado e uma entrada, um estado inicial s_0 e um conjunto de estados finais F

linguagem reconhecida por um autômato: o conjunto das seqüências de entrada que levam o estado inicial para um estado final do autômato

expressão regular: uma expressão definida recursivamente pela especificação que $\emptyset$, λ e x, para todo x no alfabeto de entrada, são expressões regulares, e que **(AB)**, **(A ∪ B)** e **A*** são expressões regulares quando **A** e **B** forem expressões regulares

conjunto regular: um conjunto definido por uma expressão regular (veja página 820)

máquina de Turing $T = (S, I, f, s_0)$: uma quádrupla que consiste em um conjunto finito S de estados, um alfabeto I que contém o símbolo branco B, uma função parcial f de $S \times I$ para $S \times I \times \{R, L\}$ e um estado inicial s_0

máquina de Turing não determinística: uma máquina de Turing que pode ter mais de uma regra de transição correspondente a cada par (estado, símbolo na fita)

problema de decisão: um problema que pergunta se afirmações de uma classe particular de afirmações são verdadeiras

problema solúvel: um problema com a propriedade que existe um algoritmo efetivo que pode resolver todos os casos do problema

problema não solúvel: um problema com a propriedade que não existe nenhum algoritmo efetivo que pode resolver todos os casos do problema

função computável: uma função cujos valores podem ser calculados usando uma máquina de Turing

função não computável: uma função cujos valores não podem ser calculados usando uma máquina de Turing

P, a classe dos problemas polinomiais no tempo: a classe dos problemas que podem ser resolvidos por uma máquina de Turing determinística em tempo polinomial em termos do tamanho da entrada

NP, a classe dos problemas polinomiais no tempo não determinísticos: a classe dos problemas que podem ser resolvidos por uma máquina de Turing não determinística em tempo polinomial em termos do tamanho da entrada

NP completo: um subconjunto da classe dos problemas NP com a propriedade que se um deles estiver na classe P, então todos os problemas em NP estarão na classe P

RESULTADOS

Para todo autômato finito não determinístico existe um autômato finito determinístico que reconhece o mesmo conjunto.

Teorema de Kleene: Um conjunto é regular se e somente se existe um autômato finito que o reconhece.

Um conjunto é regular se e somente se for gerado por uma gramática regular.

O problema de parada é não solúvel.

Questões de Revisão

1. a) Defina uma gramática de estrutura de frase.
b) O que significa uma seqüência ser derivável de uma seqüência w por uma gramática de estrutura de frase G?

2. a) O que é a linguagem gerada por uma gramática de estrutura de frase G?
b) O que é a linguagem gerada por uma gramática G com vocabulário $\{S, 0, 1\}$, conjunto de terminais $T = \{0, 1\}$, símbolo inicial S e regras de produção $S \rightarrow 000S$, $S \rightarrow 1$?
c) Dê uma gramática de estrutura de frase que gere o conjunto $\{01^n \mid n = 0, 1, 2, \ldots\}$.

3. a) Defina uma gramática de tipo 1.
b) Dê um exemplo de uma gramática que não seja uma gramática de tipo 1.
c) Defina uma gramática de tipo 2.
d) Dê um exemplo de uma gramática que não seja uma gramática de tipo 2, mas que seja uma gramática de tipo 1.

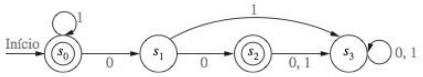
e) Defina uma gramática de tipo 3.
f) Dê um exemplo de uma gramática que não seja uma gramática de tipo 3, mas que seja uma gramática de tipo 2.

4. a) Defina uma gramática regular.
b) Defina uma linguagem regular.
c) Mostre que o conjunto $\{0^m1^n \mid m, n = 0, 1, 2, \ldots\}$ é uma linguagem regular.

5. a) O que é a forma de Backus–Naur?
b) Dê um exemplo da forma de Backus–Naur para a gramática de um subconjunto do português de sua escolha.

6. a) O que é uma máquina de estados finitos?
b) Mostre como uma máquina de vendas que aceite apenas moedas de 25 centavos e forneça um refrigerante depois de terem sido depositados 75 centavos pode ser modelada usando uma máquina de estados finitos.

7. Encontre o conjunto das seqüências reconhecidas pelo autômato finito determinístico mostrado aqui.



8. Construa um autômato finito determinístico que reconheça o conjunto das seqüências de bits que começam com 1 e terminam com 1.

9. a) O que é o fecho de Kleene de um conjunto de seqüências?
b) Encontre o fecho de Kleene do conjunto $\{11, 0\}$.

10. a) Defina um autômato finito.
b) O que significa uma seqüência ser reconhecida por um autômato finito?

11. a) Defina um autômato finito não determinístico.
b) Mostre que dado um autômato finito não determinístico, existe um autômato finito determinístico que reconhece a mesma linguagem.

12. a) Defina o conjunto das expressões regulares sobre um conjunto I.
b) Explique como as expressões regulares são usadas para representar conjuntos regulares.

13. Enuncie o Teorema de Kleene.

14. Mostre que um conjunto é gerado por uma gramática regular se e somente se ele for um conjunto regular.

15. Dê um exemplo de um conjunto que não seja reconhecido por um autômato finito. Mostre que nenhum autômato finito o reconhece.

16. Defina uma máquina de Turing.

17. Descreva como as máquinas de Turing são usadas para reconhecer conjuntos.

18. Descreva como as máquinas de Turing são usadas para calcular funções *number-theoretic*.

19. O que é um problema de decisão não solúvel? Dê um exemplo desse problema.

Exercícios Complementares

***1.** Encontre uma gramática de estrutura de frase que gere cada uma destas linguagens.
a) o conjunto das seqüências de bits da forma $0^{2n}1^{3n}$, em que n é um inteiro não negativo
b) o conjunto das seqüências de bits com o dobro de 0s do que de 1s
c) o conjunto das seqüências de bits da forma w^2, em que w é uma seqüência de bits

***2.** Encontre uma gramática de estrutura de frase que gere o conjunto $\{0^{2n} \mid n \geq 0\}$.

Para os exercícios 3 e 4, seja $G = (V, T, S, P)$ a gramática livre de contexto com $V = \{(,), S, A, B\}$, $T = \{(,)\}$, símbolo inicial S e regras de produção $S \rightarrow A$, $A \rightarrow AB$, $A \rightarrow B$, $B \rightarrow (A)$ e $B \rightarrow ()$, $S \rightarrow \lambda$.

3. Construa as árvores de derivações destas seqüências.
a) (()) **b)** ()(()) **c)** ((()()))

***4.** Mostre que $L(G)$ é o conjunto de todas as seqüências balanceadas de parênteses, definidas no preâmbulo do Exercício Complementar 55 do Capítulo 4.

Uma gramática livre de contexto é **ambígua** se existir uma palavra em $L(G)$ com duas derivações que produzem árvores de derivação diferentes, consideradas como árvores enraizadas ordenadas.

5. Mostre que a gramática $G = (V, T, S, P)$ com $V = \{0, S\}$, $T = \{0\}$, estado inicial S e regras de produção $S \rightarrow 0S$, $S \rightarrow S0$ e $S \rightarrow 0$ é ambígua, construindo duas árvores de derivação diferentes para 0^3.

6. Mostre que a gramática $G = (V, T, S, P)$ com $V = \{0, S\}$, $T = \{0\}$, estado inicial S e regras de produção $S \rightarrow 0S$ e $S \rightarrow 0$ não é ambígua.

7. Suponha que A e B sejam subconjuntos finitos de V^*, em que V é um alfabeto. É necessariamente verdade que $|AB| = |BA|$?

8. Demonstre ou mostre que é falsa cada uma destas afirmações para os subconjuntos A, B e C de V^*, em que V é um alfabeto.
a) $A(B \cup C) = AB \cup AC$
b) $A(B \cap C) = AB \cap AC$
c) $(AB)C = A(BC)$
d) $(A \cup B)^* = A^* \cup B^*$

9. Suponha que A e B sejam subconjuntos de V^*, em que V é um alfabeto. É verdade que $A \subseteq B$ se $A^* \subseteq B^*$?

10. Que conjunto de seqüências com símbolos no conjunto $\{0, 1, 2\}$ é representado pela expressão regular $(\mathbf{2}^*)(\mathbf{0} \cup (\mathbf{12}^*))^*$?

A **altura estrela** $h(E)$ de uma expressão regular no conjunto I é definida recursivamente por

$h(\emptyset) = 0$;
$h(\mathbf{x}) = 0$ se $\mathbf{x} \in I$;
$h((\mathbf{E}_1 \cup \mathbf{E}_2)) = h((\mathbf{E}_1\mathbf{E}_2)) = \max(h(\mathbf{E}_1), h(\mathbf{E}_2))$ se $\mathbf{E}_1$ e $\mathbf{E}_2$ forem expressões regulares;
$h(\mathbf{E}^*) = h(\mathbf{E}) + 1$ se $\mathbf{E}$ for uma expressão regular.

11. Encontre a altura estrela de cada uma destas expressões regulares.
a) $\mathbf{0^*1}$

b) **0*1***
c) **(0*01)***
d) **((0*1)*)***
e) **(010*)(1*01*)((01)*(10)*)***
f) **(((((0*)1)*0)*)1)***

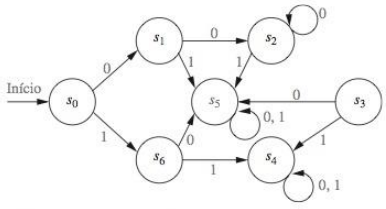
***12.** Para cada uma destas expressões regulares encontre uma expressão regular que represente a mesma linguagem com altura estrela mínima.
a) **(0*1*)***
b) **(0(01*0)*)***
c) **(0* ∪ (01)* ∪ 1*)***

13. Construa uma máquina de estados finitos com saída que produza uma saída de 1 se a seqüência de bits lida até o momento como entrada contiver quatro ou mais 1s. A seguir, construa um autômato finito determinístico que reconheça este conjunto.

14. Construa uma máquina de estados finitos com saída que produza uma saída de 1 se a seqüência de bits lida até o momento como entrada contiver quatro ou mais 1s consecutivos. A seguir, construa um autômato finito determinístico que reconheça este conjunto.

15. Construa uma máquina de estados finitos com saída que produza uma saída 1 se a seqüência de bits lida até o momento como entrada terminar com quatro ou mais 1s consecutivos. A seguir, construa um autômato finito determinístico que reconheça este conjunto.

16. Um estado s' em uma máquina de estados finitos denominado **atingível** a partir de um estado s se existir uma seqüência de entrada x tal que $f(s, x) = s'$. Um estado s é chamado de **transiente** se não existir nenhuma seqüência de entrada não vazia x com $f(s, x) = s$. Um estado s é chamado de **sorvedouro** se $f(s, x) = s$ para todas as seqüências de entrada x. Responda a estas perguntas sobre a máquina de estados finitos com o diagrama de estados ilustrado a seguir.

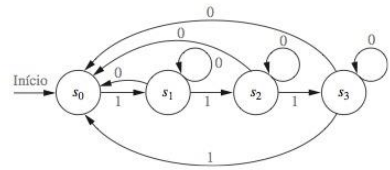


a) Quais estados são atingíveis a partir de s_0?
b) Quais estados são atingíveis a partir de s_2?
c) Quais estados são transientes?
d) Quais estados são sorvedouros?

***17.** Suponha que S, I e O são conjuntos finitos, tal que $|S| = n$, $|I| = k$ e $|O| = m$.
a) Quantas máquinas de estados finitos diferentes (máquinas de Mealy) $M = (S, I, O, f, g, s_0)$ podem ser construídas, em que o estado inicial s_0 pode ser arbitrariamente escolhido?
b) Quantas máquinas de Moore diferentes $M = (S, I, O, f, g, s_0)$ podem ser construídas, em que o estado inicial s_0 pode ser arbitrariamente escolhido?

***18.** Suponha que S e I sejam conjuntos finitos tal que $|S| = n$ e $|I| = k$. Quantos autômatos finitos diferentes $M = (S, I, f, s_0, F)$ existem, em que o estado inicial s_0 e o subconjunto F de S que consiste dos estados finais podem ser escolhidos arbitrariamente
a) se os autômatos forem determinísticos?
b) se os autômatos puderem ser não determinísticos? (*Observação*: Isso inclui os autômatos determinísticos.)

19. Construa um autômato finito determinístico que seja equivalente ao autômato não determinístico com o diagrama de estados mostrado a seguir.



20. Qual é a linguagem reconhecida pelo autômato do Exercício 19?

21. Construa autômatos finitos que reconheçam estes conjuntos.
a) **0*(10)***
b) **(01 ∪ 111)*10*(0 ∪ 1)**
c) **(001 ∪ (11)*)***

***22.** Encontre expressões regulares que representem o conjunto de todas as seqüências de 0s e 1s
a) formadas de blocos de números pares de 1s separados por um número ímpar de 0s.
b) com pelo menos dois 0s consecutivos ou três 1s consecutivos.
c) sem três zeros consecutivos ou sem dois 1s consecutivos.

***23.** Mostre que se A for um conjunto regular, então $\overline{A}$ também será.

***24.** Mostre que se A e B forem conjuntos regulares, então $A \cap B$ também será.

***25.** Encontre autômatos finitos que reconheçam estes conjuntos de seqüências de 0s e 1s.
a) o conjunto de todas as seqüências que começam com não mais de três zeros consecutivos e contêm pelo menos dois 1s consecutivos.
b) o conjunto de todas as seqüências com um número par de símbolos que não contenham o padrão 101.
c) o conjunto de todas as seqüências com pelo menos três blocos de dois ou mais 1s e pelo menos dois 0s.

***26.** Mostre que $\{0^{2n} \mid n \in \mathbf{N}\}$ não é regular. Você pode usar o lema do bombeamento dado no Exercício 22 da Seção 12.4.

***27.** Mostre que $\{1^p \mid p$ é primo$\}$ não é regular. Você pode usar o lema do bombeamento dado no Exercício 22 da Seção 12.4.

***28.** Existe um resultado para as linguagens livre de contexto análogo ao lema do bombeamento para os conjuntos regulares. Suponha que $L(G)$ seja a linguagem reconhecida por uma gramática livre de contexto G. Este resultado afirma que existe uma constante N tal que se z for uma palavra em $L(G)$ com $l(w) \geq N$, então z pode ser escrita como $uvwxy$,

em que $l(vwx) \leq N$, $l(vx) \geq 1$ e $uv^i wx^i y$ pertence a $L(G)$ para $i = 0, 1, 2, 3, \ldots$. Use este resultado para mostrar que não existe nenhuma gramática livre de contexto G com $L(G) = \{0^n 1^n 2^n \mid n = 0, 1, 2, \ldots\}$.

***29.** Construa uma máquina de Turing que calcule a função $f(n_1, n_2) = \max(n_1, n_2)$.

***30.** Construa uma máquina de Turing que calcule a função $f(n_1, n_2) = n_2 - n_1$ se $n_2 \geq n_1$ e $f(n_1, n_2) = 0$ se $n_2 < n_1$.

Desenvolvimento de Programas

(Escreva programas com as entradas e saídas abaixo.)

1. Dadas as regras de produção em uma gramática de estrutura de frase, determine qual tipo de gramática ela é no esquema de classificação de Chomsky.

2. Dadas as regras de produção em uma gramática de estrutura de frase, encontre todas as seqüências que são geradas usando vinte ou menos aplicações de suas regras de produção.

3. Dada a forma de Backus–Naur de uma gramática de tipo 2, encontre todas as seqüências que são geradas usando vinte ou menos aplicações das regras que a definem.

***4.** Dadas as regras de produção de uma linguagem livre de contexto e uma seqüência, produza uma árvore de derivação para esta seqüência se ela estiver na linguagem gerada por esta gramática.

5. Dada a tabela de estados de uma máquina de Moore e uma seqüência de entrada, produza a seqüência de saída gerada pela máquina.

6. Dada a tabela de estados de uma máquina de Mealy e uma seqüência de entrada, produza a seqüência de saída gerada pela máquina.

7. Dada a tabela de estados de um autômato finito determinístico e uma seqüência, decida se esta seqüência é reconhecida pelo autômato.

8. Dada a tabela de estados de um autômato finito não determinístico e uma seqüência, decida se esta seqüência é reconhecida pelo autômato.

***9.** Dada a tabela de estados de um autômato finito não determinístico, construa a tabela de estados de um autômato finito determinístico que reconheça a mesma linguagem.

****10.** Dada uma expressão regular, construa um autômato finito não determinístico que reconheça o conjunto que esta expressão representa.

11. Dada uma gramática regular, construa um autômato finito que reconheça a linguagem gerada por esta gramática.

12. Dado um autômato finito, construa a gramática regular que gera a linguagem reconhecida por este autômato.

***13.** Dada uma máquina de Turing, encontre a seqüência de saída produzida por uma dada seqüência de entrada.

Explorando a Computação

(Use um programa (ou programas) computacional que você escreveu para resolver estes exercícios.)

1. Resolva o problema *busy beaver* para dois estados, testando todas as máquinas de Turing possíveis com dois estados e alfabeto $\{1, B\}$.

***2.** Resolva o problema *busy beaver* para três estados testando todas as máquinas de Turing possíveis com três estados e alfabeto $\{1, B\}$.

****3.** Encontre uma máquina *busy beaver* com quatro estados testando todas as possíveis máquinas de Turing com quatro estados e alfabeto $\{1, B\}$.

****4.** Faça tanto progresso quanto puder para encontrar uma máquina *busy beaver* com cinco estados.

****5.** Faça tanto progresso quanto puder para encontrar uma máquina *busy beaver* com seis estados.

Projetos

(Responda a estas questões com informações encontradas em outras fontes.)

1. Descreva como o crescimento de certos tipos de plantas pode ser modelado usando um sistema de Lidenmeyer. Esse sistema usa uma gramática com regras de produção que modelam as diferentes maneiras pelas quais as plantas podem crescer.

2. Descreva as regras da forma de Backus–Naur (e da forma de Backus–Naur estendida) usadas para especificar a sintaxe de uma linguagem de programação, tal como Java, LISP ou ADA ou a linguagem de banco de dados SQL.

3. Explique como as máquinas de estados finitos são usadas por corretores ortográficos.
4. Explique como as máquinas de estados finitos são usadas no estudo de protocolos de redes.
5. Explique como as máquinas de estados finitos são usadas em programas de reconhecimento de fala.
6. Explique o conceito de minimização de autômato finito. Dê um algoritmo que execute esta minimização.
7. Dê a definição de autômato celular. Explique suas aplicações. Use o Jogo da Vida como um exemplo.
8 Defina um autômato de pilha. Explique como os autômatos de pilha são usados para reconhecer conjuntos. Quais conjuntos são reconhecidos pelos autômatos de pilhas? Forneça um esboço de uma demonstração que justifique sua resposta.
9. Defina um autômato linearmente limitado. Explique como os autômatos linearmente limitados são usados para reconhecer conjuntos. Quais conjuntos são reconhecidos pelos autômatos linearmente limitados? Forneça um esboço de uma demonstração que justifique sua resposta.
10. Procure a definição original de Turing do que chamamos agora de máquina de Turing. Qual a motivação dele para definir estas máquinas?
11. Descreva o conceito de máquina de Turing universal. Explique como essa máquina pode ser construída.
12. Explique os tipos de aplicações nas quais as máquinas de Turing não determinísticas são usadas em vez das máquinas de Turing determinísticas.
13. Mostre que uma máquina de Turing pode simular qualquer ação de uma máquina de Turing não determinística.
14. Mostre que um conjunto é reconhecido por uma máquina de Turing se e somente se ele é gerado por uma gramática de estrutura de frase.
15. Descreva os conceitos básicos do cálculo lambda e explique como ele é usado para estudar a computabilidade de funções.
16. Mostre que uma máquina de Turing, como definida neste capítulo, pode fazer qualquer coisa que uma máquina com n fitas pode fazer.
17. Mostre que uma máquina de Turing com uma fita infinita em uma direção pode fazer qualquer coisa que uma máquina de Turing com uma fita infinita em ambas as direções pode fazer.

APÊNDICE

A-1 Axiomas para os Números Reais e para os Inteiros Positivos

Neste livro supusemos um conjunto explícito de axiomas para o conjunto dos números reais e para o conjunto dos inteiros positivos. Neste apêndice listaremos esses axiomas e ilustraremos como fatos básicos, também usados sem demonstração no texto, podem ser deduzidos usando-os.

Axiomas para os Números Reais

Os axiomas-padrão para os números reais incluem tanto os **axiomas de corpo** (ou **algébricos**), usados para especificar regras para as operações aritméticas básicas, quanto os **axiomas de ordem**, usados para especificar as propriedades de ordem dos números reais.

AXIOMAS DE CORPO Começamos com os axiomas de corpo. Como é usual, indicaremos a soma e o produto de dois números reais x e y como $x + y$ e $x \cdot y$, respectivamente. (Observe que o produto de x e y é freqüentemente indicado por xy sem o uso do ponto para indicar a multiplicação. Não usaremos esta notação abreviada neste apêndice, mas a usaremos no texto principal.) Além disso, por convenção, faremos multiplicações antes das adições, a menos que sejam usados parênteses. Embora estas afirmações sejam axiomas, elas são comumente chamadas de *propriedades* ou *regras*. Os dois primeiros destes axiomas nos dizem que, quando somamos ou multiplicamos dois números reais, o resultado é de novo um número real; estas são as *propriedades de fechamento*.

Propriedade de Fechamento para a Adição Para todos os números reais x e y, $x + y$ é um número real.
Propriedade de Fechamento para a Multiplicação Para todos os números reais x e y, $x \cdot y$ é um número real.

Os próximos dois axiomas nos dizem que, quando somamos ou multiplicamos três números reais, obtemos o mesmo resultado independentemente da ordem das operações; estas são as *propriedades associativas*.

Propriedade Associativa da Adição Para todos os números reais x, y e z, $(x + y) + z = x + (y + z)$.
Propriedade Associativa da Multiplicação Para todos os números reais x, y e z, $(x \cdot y) \cdot z = x \cdot (y \cdot z)$.

Dois axiomas algébricos adicionais nos dizem que a ordem na qual somamos ou multiplicamos dois números não importa; estas são as *propriedades comutativas*.

Propriedade Comutativa da Adição Para todos os números reais x e y, $x + y = y + x$.
Propriedade Comutativa da Multiplicação Para todos os números reais x e y, $x \cdot y = y \cdot x$.

Os próximos dois axiomas nos dizem que 0 e 1 são elementos neutros da adição e da multiplicação no conjunto dos números reais. Ou seja, quando somamos 0 a qualquer número real ou multiplicamos um número real por 1 não mudamos este número real. Estas propriedades são chamadas de *propriedades dos elementos neutros*.

Propriedade do Elemento Neutro da Adição Para todo número real x, $x + 0 = 0 + x = x$.
Propriedade do Elemento Neutro da Multiplicação Para todo número real x, $x \cdot 1 = 1 \cdot x = x$.

Embora pareça óbvio, precisamos também do seguinte axioma.

Propriedade dos Elementos Neutros O elemento neutro da adição, 0, e o elemento neutro da multiplicação, 1, são distintos, ou seja, $0 \neq 1$.

Dois axiomas adicionais nos dizem que, para todo número real, existe um número real que pode ser somado a este número para produzir 0, e, para todo número real não nulo, existe um

número real pelo qual podemos multiplicá-lo para produzir 1. Estas são as *propriedades dos elementos inversos*.

Propriedade do Elemento Inverso da Adição Para todo número real x, existe um número real $-x$ (chamado de *inverso aditivo* de x) tal que $x + (-x) = (-x) + x = 0$.

Propriedade do Elemento Inverso da Multiplicação Para todo número real x não nulo, existe um número real $1/x$ (chamado de *inverso multiplicativo* de x) tal que $x \cdot (1/x) = (1/x) \cdot x = 1$.

Os últimos axiomas algébricos para os números reais são as *propriedades distributivas*, que nos dizem que a multiplicação se distribui sobre somas; ou seja, que obtemos o mesmo resultado quando primeiro somamos um par de números reais e depois o multiplicamos por um terceiro número real ou quando multiplicamos cada um desses dois números reais pelo terceiro número real e depois somamos estes dois produtos.

Propriedades Distributivas Para todos os números reais x, y e z, $x \cdot (y + z) = x \cdot y + x \cdot z$ e $(x + y) \cdot z = x \cdot z + y \cdot z$.

AXIOMAS DE ORDEM A seguir, enunciaremos os *axiomas de ordem* para os números reais, que especificam propriedades da relação "maior que", indicada por $>$, no conjunto dos números reais. Escrevemos $x > y$ (e $y < x$) quando x for maior que y e escrevemos $x \geq y$ (e $y \leq x$) quando $x > y$ ou $x = y$. O primeiro desses axiomas nos diz que, dados dois números reais, exatamente uma entre três possibilidades ocorre: os dois números são iguais, o primeiro é maior que o segundo ou o segundo é maior que o primeiro. Esta regra é chamada de *propriedade da tricotomia*.

Propriedade da Tricotomia Para todos os números reais x e y, exatamente uma entre $x = y$, $x > y$ ou $y > x$ é verdadeira.

A seguir, temos um axioma, chamado de *propriedade transitiva*, que nos diz que, se um número for maior que um segundo número e este segundo número for maior que um terceiro, então o primeiro número é maior que o terceiro.

Propriedade Transitiva Para todos os números reais x, y e z, se $x > y$ e $y > z$, então $x > z$.

Temos também duas *propriedades de compatibilidade*, que nos dizem que, quando somamos um número a ambos os lados de uma relação maior que, a relação maior que é preservada e, quando multiplicamos ambos os lados de uma relação maior que por um *número real positivo* (isto é, por um número real x com $x > 0$), a relação maior que é preservada.

Propriedade de Compatibilidade da Adição Para todos os números reais x, y e z, se $x > y$, então $x + z > y + z$.

Propriedade de Compatibilidade da Multiplicação Para todos os números reais x, y e z, se $x > y$ e $z > 0$, então $x \cdot z > y \cdot z$.

Deixamos para o leitor demonstrar que, para todos os números reais x, y e z, se $x > y$ e $z < 0$, então $x \cdot z < y \cdot z$ (veja o Exercício 15). Ou seja, a multiplicação de uma desigualdade por um número real negativo inverte o sentido da desigualdade.

O último axioma para o conjunto dos números reais é a *propriedade da completude*. Antes de enunciar este axioma, precisamos de algumas definições. Primeiro, dado um conjunto não vazio A de números reais, dizemos que o número real b é um **limitante superior** de A se, para todo número real a em A, $b \geq a$. Um número real s é o **menor limitante superior** (ou **supremo**) de A se s for um limitante superior de A e se, sempre que t for um limitante superior de A, tivermos $s \leq t$.

Propriedade da Completude Todo conjunto não vazio de números reais que seja limitado superiormente tem um menor limitante superior.

Usando os Axiomas para Demonstrar Fatos Básicos

Os axiomas que listamos podem ser usados para demonstrar muitas propriedades que usamos freqüentemente sem menção explícita. Damos diversos exemplos de resultados que podemos

demonstrar usando os axiomas e deixamos a demonstração de diversas outras propriedades como exercícios. Embora os resultados que demonstraremos pareçam bastante óbvios, demonstrá-los usando os axiomas que enunciamos pode ser desafiador.

TEOREMA 1 O elemento neutro 0 da adição dos números reais é único.

Demonstração: Para mostrar que o elemento neutro 0 da adição dos números reais é único, suponha que $0'$ também seja um elemento neutro para a adição de números reais. Isto significa que $0' + x = x + 0' = x$ sempre que x seja um número real. Pela propriedade do elemento neutro da adição, segue que $0 + 0' = 0'$. Como $0'$ é um elemento neutro da adição, sabemos que $0 + 0' = 0$. Segue que $0 = 0'$, pois ambos são iguais a $0 + 0'$. Isto mostra que 0 é o único elemento neutro da adição dos números reais. ◁

TEOREMA 2 O inverso da adição de um número real x é único.

Demonstração: Seja x um número real. Suponha que y e z sejam ambos inversos aditivos de x. Então,

$$
\begin{aligned}
y &= 0 + y && \text{pela propriedade do elemento neutro da adição}\\
&= (z + x) + y && \text{pois } z \text{ é um inverso aditivo de } x\\
&= z + (x + y) && \text{pela propriedade associativa da adição}\\
&= z + 0 && \text{pois } y \text{ é um inverso aditivo de } x\\
&= z && \text{pela propriedade do elemento neutro da adição}
\end{aligned}
$$

Segue que $y = z$. ◁

Os teoremas 1 e 2 nos dizem que o elemento neutro da adição e o inverso aditivo são únicos. Os teoremas 3 e 4 nos dizem que o elemento neutro da multiplicação e os inversos multiplicativos de números reais não nulos também são únicos. Deixamos suas demonstrações como exercícios no final deste apêndice.

TEOREMA 3 O elemento neutro 1 da multiplicação dos números reais é único.

TEOREMA 4 O inverso da multiplicação de um número real x não nulo é único.

TEOREMA 5 Para todo número real x, $x \cdot 0 = 0$.

Demonstração: Suponha que x seja um número real. Pela propriedade do elemento inverso da adição, existe um número real y tal que ele é o inverso aditivo de $x \cdot 0$, de modo que temos $x \cdot 0 + y = 0$. Pela propriedade do elemento neutro da adição, $0 + 0 = 0$. Usando a propriedade distributiva, vemos que $x \cdot 0 = x \cdot (0 + 0) = x \cdot 0 + x \cdot 0$. Segue que

$$0 = x \cdot 0 + y = (x \cdot 0 + x \cdot 0) + y.$$

A seguir, observe que, pela propriedade associativa da adição e pelo fato de $x \cdot 0 + y = 0$, temos

$$(x \cdot 0 + x \cdot 0) + y = x \cdot 0 + (x \cdot 0 + y) = x \cdot 0 + 0.$$

Finalmente, pela propriedade do elemento neutro da adição, sabemos que $x \cdot 0 + 0 = x \cdot 0$. Conseqüentemente, $x \cdot 0 = 0$. ◁

TEOREMA 6 Para todos os números reais x e y, se $x \cdot y = 0$, então $x = 0$ ou $y = 0$.

Demonstração: Suponha que x e y sejam números reais e que $x \cdot y = 0$. Se $x \neq 0$, então, pela propriedade do elemento inverso da multiplicação, x tem um inverso multiplicativo $1/x$, tal que $x \cdot (1/x) = (1/x) \cdot x = 1$. Como $x \cdot y = 0$, temos $(1/x) \cdot (x \cdot y) = (1/x) \cdot 0 = 0$ pelo Teorema 5. Usando a propriedade associativa da multiplicação, temos $((1/x) \cdot x) \cdot y = 0$. Isto significa que $1 \cdot y = 0$. Pela propriedade do elemento neutro da multiplicação, vemos que $1 \cdot y = y$, de modo que $y = 0$. Conseqüentemente, ou $x = 0$ ou $y = 0$. ◁

TEOREMA 7 O elemento neutro 1 da multiplicação no conjunto dos números reais é maior que o elemento neutro 0 da adição.

Demonstração: Pela propriedade da tricotomia, ou $0 = 1$, ou $0 > 1$, ou $1 > 0$. Sabemos pela propriedade dos elementos neutros que $0 \neq 1$.

Assim, suponha que $0 > 1$. Mostraremos que esta hipótese nos leva a uma contradição. Pela propriedade do elemento inverso da adição, 1 tem um inverso aditivo -1, com $1 + (-1) = 0$. A propriedade de compatibilidade da adição nos diz que $0 + (-1) > 1 + (-1) = 0$; a propriedade do elemento neutro da adição nos diz que $0 + (-1) = -1$. Conseqüentemente, $-1 > 0$, e pela propriedade de compatibilidade da multiplicação, $(-1) \cdot (-1) > (-1) \cdot 0$. Pelo Teorema 5, o lado direito da última desigualdade é 0. Pela propriedade distributiva, $(-1) \cdot (-1) + (-1) \cdot 1 = (-1) \cdot (-1 + 1) = (-1) \cdot 0 = 0$. Portanto, o lado esquerdo desta última desigualdade, $(-1) \cdot (-1)$, é o inverso aditivo único de -1, de modo que este lado da desigualdade é igual a 1. Conseqüentemente, esta última desigualdade se torna $1 > 0$, contradizendo a propriedade da tricotomia, já que tínhamos suposto que $0 > 1$.

Como sabemos que $0 \neq 1$ e que é impossível $0 > 1$, pela propriedade da tricotomia, concluímos que $1 > 0$. ◁

Links

ARQUIMEDES (287 a.C.–212 a.C.) Arquimedes foi um dos maiores cientistas e matemáticos da antiguidade. Ele nasceu em Siracusa, uma cidade-estado grega na Sicília. Seu pai, Fídias, foi um astrônomo. Arquimedes foi educado em Alexandria, no Egito. Depois de completar seus estudos, ele voltou para Siracusa, onde passou o resto de sua vida. Pouco se sabe sobre sua vida pessoal; não sabemos se ele se casou ou se teve filhos. Arquimedes foi morto em 212 a.C. por um soldado romano quando os romanos tomaram Siracusa.

Arquimedes fez muitas descobertas importantes em geometria. Seu método para calcular a área sob uma curva foi descrito dois mil anos antes de suas idéias serem reinventadas como parte do cálculo integral. Arquimedes também desenvolveu um método para expressar inteiros grandes que não podiam ser expressos pelos métodos gregos usuais. Ele descobriu um método para calcular o volume de uma esfera, bem como de outros sólidos, e calculou uma aproximação do π. Arquimedes também foi engenheiro e inventor de sucesso; sua máquina para bombear água, chamada agora de *bomba de Arquimedes*, ainda é usada nos dias de hoje. Talvez sua descoberta mais conhecida seja o *princípio de flutuação*, que nos diz que um objeto submerso em líquido se torna mais leve pela ação de uma quantidade igual ao peso que ele desloca. Algumas histórias nos dizem que Arquimedes foi precursor da prática dos streakers (nudistas), correndo nu pelas ruas de Siracusa gritando "Eureca" (que quer dizer "eu descobri") quando fez sua descoberta. Ele também é conhecido pelo seu uso inteligente das máquinas que detiveram as forças romanas que atacaram Siracusa por vários anos durante a segunda guerra Púnica.

O próximo teorema nos diz que para todo número real existe um inteiro (em que *inteiro* quer dizer zero, a soma de qualquer número de 1s e os inversos aditivos dessas somas) maior que este número real. Este resultado é atribuído ao matemático grego Arquimedes e pode ser encontrado no Livro V dos *Elementos* de Euclides.

TEOREMA 8 **PROPRIEDADE ARQUIMEDIANA** Para todo número real x existe um inteiro n tal que $n > x$.

Demonstração: Suponha que x seja um número real com $n \leq x$ para todo inteiro n. Então, x é um limitante superior para o conjunto dos inteiros. Pela propriedade da completude, segue que o conjunto dos inteiros tem um menor limitante superior M. Como $M - 1 < M$ e M é um menor limitante superior do conjunto dos inteiros, $M - 1$ não é um limitante superior do conjunto dos inteiros. Isto significa que existe um inteiro n com $n > M - 1$. Isto implica que $n + 1 > M$, o que contradiz o fato de M ser um limitante superior do conjunto dos inteiros. ◁

Axiomas para o Conjunto dos Inteiros Positivos

Os axiomas que listamos agora especificam o conjunto dos inteiros positivos como um subconjunto dos inteiros que satisfazem quatro propriedades-chave. Supomos a veracidade destes axiomas neste livro.

Axioma 1 O número 1 é um inteiro positivo.

Axioma 2 Se n for um inteiro positivo, então $n + 1$, o *sucessor* de n, também é um inteiro positivo.

Axioma 3 Todo inteiro positivo diferente de 1 é o sucessor de um inteiro positivo.

Axioma 4 **O Princípio da Boa Ordenação** Todo subconjunto não vazio do conjunto dos inteiros positivos tem um menor elemento.

Nas seções 4.1 e 4.2 é mostrado que o princípio da boa ordem é equivalente ao princípio da indução matemática.

Axioma da Indução Matemática Se S for um subconjunto dos inteiros positivos, tal que $1 \in S$ e se $n \in S$, então $n + 1 \in S$, então S é o conjunto dos inteiros positivos.

A maioria dos matemáticos considera o sistema dos números reais como já existente, com os números reais satisfazendo os axiomas que listamos neste apêndice. Entretanto, os matemáticos do século XIX desenvolveram técnicas para construir o conjunto dos números reais, começando com conjuntos mais básicos de números. (O processo de construção dos números reais às vezes é estudado em cursos de graduação de matemática mais avançados. Um tratamento disto pode ser encontrado em [Mo87], por exemplo.) O primeiro passo no processo é a construção do conjunto dos inteiros positivos usando os Axiomas 1 a 3 e ou o Princípio da Boa Ordem ou o Axioma de Indução Matemática. Então, são definidas as operações de adição e multiplicação de inteiros positivos. Uma vez que isto tenha sido feito, o conjunto dos inteiros pode ser construído usando classes de equivalência de pares de inteiros positivos em que $(a, b) \sim (c, d)$ se e somente se $a + d = b + c$; a adição e a multiplicação de inteiros podem ser definidas usando estes pares (veja o Exercício 21). (Relações de equivalência e classes de equivalência foram discutidas no Capítulo 8.) A seguir, o conjunto dos números racionais pode ser construído usando as classes de equivalência de pares de inteiros em que o segundo inteiro no par não é zero, em que $(a, b) \approx (c, d)$ se e somente se $a \cdot d = b \cdot c$; a adição e a multiplicação do conjunto dos números racionais podem ser definidas em termos desses pares (veja o Exercício 22). Usando seqüências infinitas, o conjunto dos números reais pode ser construído a partir do conjunto dos números racionais. O leitor interessado achará que vale a pena ler os muitos detalhes dos passos desta construção.

Exercícios

Use apenas os axiomas e os teoremas deste apêndice nas demonstrações de suas respostas a estes exercícios.

1. Demonstre o Teorema 3, que afirma que o elemento neutro da multiplicação dos números reais é único.
2. Demonstre o Teorema 4, que afirma que, para todo real não nulo x, o elemento inverso multiplicativo de x é único.
3. Demonstre que para todos os números reais x e y, $(-x) \cdot y = x \cdot (-y) = -(x \cdot y)$.
4. Demonstre que para todos os números reais x e y, $-(x + y) = (-x) + (-y)$.
5. Demonstre que para todos os números reais x e y, $(-x) \cdot (-y) = x \cdot y$.
6. Demonstre que, para todos os números reais x, y e z, se $x + z = y + z$, então $x = y$.
7. Demonstre que, para todo número real x, $-(-x) = x$.

Defina a **diferença** $x - y$ dos números reais x e y por $x - y = x + (-y)$, em que $-y$ é o inverso aditivo de y, e o **quociente** x/y, em que $y \neq 0$, por $x/y = x \cdot (1/y)$, em que $1/y$ é o inverso multiplicativo de y.

8. Demonstre que, para todos os números reais x e y, $x = y$ se e somente se $x - y = 0$.
9. Demonstre que, para todos os números reais x e y, $-x - y = -(x + y)$.
10. Demonstre que, para todos os números reais não nulos x e y, $1/(x/y) = y/x$, em que $1/(x/y)$ é o inverso multiplicativo de x/y.
11. Demonstre que, para todos os números reais w, x, y e z, se $x \neq 0$ e $z \neq 0$, então $(w/x) + (y/z) = (w \cdot z + x \cdot y)/(x \cdot z)$.
12. Demonstre que, para todo número real positivo x, $1/x$ também será um número real positivo.
13. Demonstre que, para todos os números reais positivos x e y, $x \cdot y$ também será um número real positivo.
14. Demonstre que, para todos os números reais x e y, se $x > 0$ e $y < 0$, então $x \cdot y < 0$.
15. Demonstre que, para todos os números reais x, y e z, se $x > y$ e $z < 0$, então $x \cdot z < y \cdot z$.
16. Demonstre que, para todo número real x, $x \neq 0$ se e somente se $x^2 > 0$.
17. Demonstre que, para todos os números reais w, x, y e z, se $w < x$ e $y < z$, então $w + y < x + z$.
18. Demonstre que, para todos os números reais positivos x e y, se $x < y$, então $1/x > 1/y$.
19. Demonstre que, para todo número real positivo x, existe um inteiro positivo n tal que $n \cdot x > 1$.

*20. Demonstre que, entre dois números reais distintos existe um número racional (isto é, um número da forma x/y, em que x e y são inteiros com $y \neq 0$).

Os exercícios 21 e 22 envolvem a noção de relação de equivalência, discutida no Capítulo 8 do texto.

*21. Defina a relação $\sim$ no conjunto de pares ordenados de inteiros positivos por $(w, x) \sim (y, z)$ se e somente se $w + z = x + y$. Mostre que as operações $[(w, x)]_\sim + [(y, z)]_\sim = [(w + y, x + z)]_\sim$ e $[(w, x)]_\sim \cdot [(y, z)]_\sim = [(w \cdot y + x \cdot z, x \cdot y + w \cdot z)]_\sim$ estão bem definidas, ou seja, elas não dependem do representante das classes de equivalência escolhido para o cálculo.

*22. Defina a relação $\approx$ nos pares ordenados de inteiros com o segundo elemento não nulo por $(w, x) \approx (y, z)$ se e somente se $w \cdot z = x \cdot y$. Mostre que as operações $[(w, x)]_\approx + [(y, z)]_\approx = [(w \cdot z + x \cdot y, x \cdot z)]_\approx$ e $[(w, x)]_\approx \cdot [(y, z)]_\approx = [(w \cdot y, x \cdot z)]_\approx$ estão bem definidas, ou seja, elas não dependem do representante das classes de equivalência escolhido para o cálculo.

APÊNDICE

A-2 Funções Exponenciais e Logarítmicas

Neste apêndice revisamos algumas das propriedades básicas das funções exponenciais e dos logaritmos. Essas propriedades foram usadas por todo o texto. Os estudantes que precisem de mais revisão deste material devem consultar livros de pré-cálculo ou de cálculo, como os mencionados nas Sugestões de Leitura.

Funções Exponenciais

Seja n um inteiro positivo e seja b um número real positivo fixo. A função $f_b(n) = b^n$ é definida por

$$f_b(n) = b^n = b \cdot b \cdot b \cdot \cdots \cdot b,$$

em que existem n fatores de b multiplicados no lado direito da equação.

Definimos a função $f_b(x) = b^x$ para todos os números reais x usando técnicas do cálculo. A função $f_b(x) = b^x$ é chamada de **função exponencial de base *b***. Não discutiremos como encontrar os valores das funções exponenciais de base b quando x não for um inteiro.

Duas das propriedades importantes satisfeitas pelas funções exponenciais são dadas no Teorema 1. Demonstrações delas e de outras propriedades relacionadas podem ser encontradas em textos de cálculo.

TEOREMA 1 Seja b um número real positivo. Então

1. $b^{x+y} = b^x b^y$, e
2. $(b^x)^y = b^{xy}$.

Mostramos os gráficos de algumas funções exponenciais na Figura 1.

Funções Logarítmicas

Suponha que b seja um número real com $b > 1$. Então, a função exponencial b^x é estritamente crescente (um fato mostrado no cálculo). Ela é uma correspondência biunívoca do conjunto dos números reais com o conjunto dos números reais positivos. Logo, esta função tem uma inversa, chamada de **função logarítmica de base *b***. Em outras palavras, se b for um número real maior que 1 e x for um número real positivo, então

$$b^{\log_b x} = x.$$

O valor desta função em x é chamado de **logaritmo de *x* na base *b***.

Da definição, segue que

$$\log_b b^x = x.$$

Damos diversas propriedades importantes do logaritmo no Teorema 2.

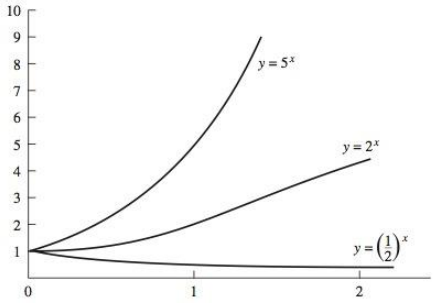


FIGURA 1 Gráficos das Funções Exponenciais de Bases $\frac{1}{2}$, 2 e 5.

TEOREMA 2 Seja b um número real maior que 1. Então

1. $\log_b(xy) = \log_b x + \log_b y$, sempre que x e y forem números reais positivos, e
2. $\log_b(x^y) = y \log_b x$ sempre que x for um número real positivo.

Demonstração: Como $\log_b(x\,y)$ é o único número real tal que $b^{\log_b(x\,y)} = xy$, para demonstrar a parte 1 basta mostrar que $b^{\log_b x + \log_b y} = xy$. Pela parte 1 do Teorema 1, temos

$$\begin{aligned} b^{\log_b x + \log_b y} &= b^{\log_b x} b^{\log_b y} \\ &= xy. \end{aligned}$$

Para demonstrar a parte 2, basta mostrar que $b^{y \log_b x} = x^y$. Pela parte 2 do Teorema 1, temos

$$\begin{aligned} b^{y \log_b x} &= (b^{\log_b x})^y \\ &= x^y. \end{aligned}$$

◁

O teorema a seguir relaciona logaritmos em duas bases diferentes.

TEOREMA 3 Sejam a e b dois números reais maiores que 1, e seja x um número real positivo. Então,

$$\log_a x = \log_b x / \log_b a.$$

Demonstração: Para demonstrar este resultado, basta mostrar que

$$b^{\log_a x \cdot \log_b a} = x.$$

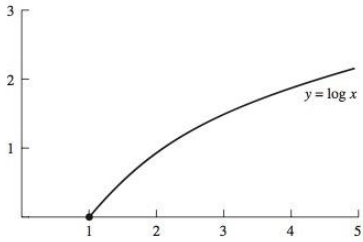


FIGURA 2 O Gráfico de $f(x) = \log x$.

Pela parte 2 do Teorema 1, temos

$$\begin{aligned} b^{\log_a x \cdot \log_b a} &= (b^{\log_b a})^{\log_a x} \\ &= a^{\log_a x} \\ &= x. \end{aligned}$$

Isso completa a demonstração. ◁

Como a base mais freqüentemente usada para logaritmos neste texto é $b = 2$, a notação $\log x$ é usada por todo o texto para indicar $\log_2 x$.

O gráfico da função $f(x) = \log x$ é mostrado na Figura 2. Do Teorema 3, quando for usada uma base b diferente de 2, é obtida uma função que é a multiplicação de $\log x$ por uma constante, a saber, $(1/\log b)\log x$.

Exercícios

1. Expresse cada uma das seguintes quantidades como potências de 2.
a) $2 \cdot 2^2$ **b)** $(2^2)^3$ **c)** $2^{(2^2)}$

2. Encontre cada uma das seguintes quantidades.
a) $\log_2 1024$ **b)** $\log_2 1/4$ **c)** $\log_4 8$

3. Suponha que $\log_4 x = y$. Encontre cada uma das seguintes quantidades.
a) $\log_2 x$ **b)** $\log_8 x$ **c)** $\log_{16} x$

☞ **4.** Sejam a, b e c números reais positivos. Mostre que $a^{\log_b c} = c^{\log_b a}$.

5. Desenhe o gráfico de $f(x) = b^x$ se b for
a) 3.
b) $1/3$.
c) 1.

6. Desenhe o gráfico de $f(x) = \log_b x$ se b for
a) 4. **b)** 100. **c)** 1 000.

APÊNDICE

A-3 Pseudocódigo

Os algoritmos neste texto foram descritos tanto em português quanto em **pseudocódigo**. Pseudocódigo é uma etapa intermediária entre a descrição na língua portuguesa dos passos de um procedimento e uma especificação deste procedimento usando uma linguagem de programação real. As vantagens de usar um pseudocódigo incluem a simplicidade com a qual ele pode ser escrito e entendido e a facilidade de produzir códigos computacionais reais (em diversas linguagens de programação) a partir do pseudocódigo. Descreveremos os tipos particulares de **comandos**, ou instruções de alto nível, dos pseudocódigos que usaremos. Cada um desses comandos em pseudocódigo pode ser traduzido em um ou mais comandos em uma linguagem de programação particular, o que por sua vez pode ser traduzido em uma ou mais (possivelmente muitas) instruções de baixo nível para o computador.

Links

Este apêndice descreve a forma e a sintaxe do pseudocódigo usado no texto. Este pseudocódigo é planejado de modo que sua estrutura básica seja parecida com a de Pascal, que atualmente é a linguagem de programação mais ensinada. Entretanto, o pseudocódigo que usaremos será muito mais livre que uma linguagem de programação formal, pois muito da descrição em língua portuguesa dos passos será permitida.

Este apêndice não tem a intenção de ser um estudo formal. Em vez disto, ele deve servir como um guia de referência para os estudantes quando estudarem as descrições dos algoritmos dadas no texto e quando escreverem soluções em pseudocódigo para os exercícios.

Comandos de Procedimento

O pseudocódigo para um algoritmo começa com um comando de **procedimento** que dá o nome de um algoritmo, lista as variáveis de entrada e descreve que tipo de variável é cada entrada. Por exemplo, o comando

procedure *maximo*(L: lista de inteiros)

é o primeiro comando na descrição em pseudocódigo do algoritmo, ao qual demos o nome de *maximo*, que encontra o máximo de uma lista de *L* inteiros.

Comandos de Atribuição de Valores e de Outros Tipos

Um comando de atribuição de valores é usado para atribuir valores às variáveis. Em um comando de atribuição de valores o lado esquerdo é o nome da variável e o lado direito é uma expressão que envolve constantes, variáveis para as quais já foram atribuídos valores ou funções definidas por procedimentos. O lado direito pode conter qualquer uma das operações aritméticas usuais. Entretanto, nos pseudocódigos deste livro ele pode incluir qualquer operação bem definida, mesmo se esta operação puder ser executada usando-se apenas um número maior de comandos em uma linguagem de programação real.

O símbolo := é usado para atribuição de valores. Assim, um comando de atribuição de valores tem a forma

variável := *expressão*

Por exemplo, o comando

```
max := a
```

atribui o valor *a* à variável *max*. Um comando como

```
x := o maior inteiro na lista L
```

também pode ser usado. Isso iguala *x* ao maior inteiro na lista *L*. Traduzir este comando para uma linguagem de programação real exigiria mais de um comando. Além disso, o comando

```
troque a e b
```

pode ser usado para trocar *a* e *b*. Poderíamos também expressar este comando com diversos comandos de atribuição de valores (veja o Exercício 2 no final deste apêndice), mas, por simplicidade, em geral preferiremos esta forma abreviada de pseudocódigo.

Blocos de Comandos

Comandos podem ser agrupados em blocos para executar procedimentos complicados. Estes blocos são demarcados usando comandos **begin** e **end**, e os comandos no bloco são todos endentados do mesmo tanto.

```
begin
    comando 1
    comando 2
    comando 3
    .
    .
    .
    comando n
end
```

Os comandos no bloco são executados sequencialmente.

Comentários

Nos pseudocódigos deste livro, comandos delimitados por chaves não são executados. Esses comandos servem como comentários ou lembretes que ajudam a explicar como o procedimento funciona. Por exemplo, o comando

{x é o maior elemento em L}

pode ser usado para lembrar o leitor que neste ponto do procedimento a variável x é igual ao maior elemento na lista L.

Construções Condicionais

A forma mais simples de uma construção condicional que usaremos é

if condição **then** comando

ou

if condição **then**
begin
    bloco de comandos
end

Aqui, a condição é verificada, e se ela for verdadeira, então o comando é executado. Por exemplo, no Algoritmo 1 da Seção 3.1, que encontra o máximo de um conjunto de inteiros, usamos um comando condicional para verificar se $max < a_i$ para cada variável; se for, atribuímos o valor de a_i à max.

Freqüentemente, precisamos usar um tipo mais geral de construção. Isso é usado quando queremos fazer uma coisa quando a condição indicada for verdadeira, mas uma outra quando for falsa. Usamos a construção

if condição **then** comando 1
else comando 2

Observe que um dos ou ambos os comandos 1 e 2 podem ser substituídos por um bloco de comandos. Às vezes, precisamos usar uma forma ainda mais geral de condicional. A forma geral de construção condicional que usaremos é

if condição 1 **then** comando 1
else if condição 2 **then** comando 2
else if condição 3 **then** comando 3
    .
    .
    .
else if condição n **then** comando n
else comando $n + 1$

Quando esta construção é usada, se a condição 1 for verdadeira, então o comando 1 é executado e o programa sai da construção. Além disso, se a condição 1 for falsa, o programa verifica se a condição 2 é verdadeira; se for, o comando 2 é executado, e assim por diante. Logo, se nenhuma das $n - 1$ primeiras condições for válida, mas a condição n for, o comando n é executado. Finalmente, se nenhuma das condições 1, 2, 3, ... n for verdadeira, então o comando $n + 1$ é executado. Observe que qualquer um dos $n + 1$ comandos pode ser substituído por um bloco de comandos.

Construções de Laços

Existem dois tipos de construções de laços no pseudocódigo deste livro. O primeiro é a construção "for", que tem a forma

```
for variavel := valor inicial to valor final
    comando
```

ou

```
for variavel := valor inicial to valor final
begin
    bloco de comandos
end
```

Aqui, no início do laço, é atribuído o valor *valor inicial* à *variavel* se *valor inicial* for menor que ou igual a *valor final*, e os comandos no final desta construção são executados com este valor de *variavel*. Então, *variavel* é aumentado por um e o comando, ou comandos no bloco, são executados com este novo valor de *variavel*. Isto é repetido até que *variavel* atinja *valor final*. Depois das instruções serem executadas com *variavel* igual a *valor final*, o algoritmo prossegue para o próximo comando. Quando o *valor inicial* exceder o *valor final*, nenhum dos comandos do laço é executado.

Podemos usar a construção de laço "for" para encontrar a soma dos inteiros positivos de 1 a n com o seguinte pseudocódigo.

```
soma := 0
for i := 1 to n
    soma := soma + i
```

Além disso, o comando "for" mais geral, da forma

```
for todos os elementos com certa propriedade
```

é usado neste texto. Isso significa que o comando ou bloco de comandos que vem a seguir é executado sucessivamente para os elementos com a propriedade dada.

O segundo tipo de construção de laço que usaremos é a construção "while". Isso tem a forma

```
while condição
    comando
```

ou

```
while condição
begin
    bloco de comandos
end
```

Quando esta condição é usada, a condição dada é verificada e, se for verdadeira, os comandos que vêm a seguir são executados, o que pode mudar os valores das variáveis que fazem parte da condição. Se a condição ainda for verdadeira depois destas instruções terem sido executadas, as instruções são executadas novamente. Isso é repetido até que a condição se torne falsa. Como exemplo, podemos encontrar a soma dos inteiros de 1 a n usando o seguinte bloco de pseudocódigo incluindo uma construção "while"

```
soma := 0
while n > 0
begin
    soma := soma + n
    n := n − 1
end
```

Observe que qualquer construção "for" pode ser transformada em uma construção "while" (veja o Exercício 3 no final deste apêndice). Entretanto, muitas vezes é mais fácil entender a construção "for". Assim, quando fizer sentido, usaremos a construção "for" com preferência à construção "while" correspondente.

Laços dentro de Laços

Laços ou comandos condicionais são usados com freqüência dentro de outros laços ou comandos condicionais. No pseudocódigo deste livro, quando um laço ou um comando condicional é o conteúdo integral de um laço ou de uma construção condicional, nem sempre usaremos um par separado de comandos **begin** e **end** para delimitar os comandos que são executados durante cada iteração ou quando uma condição especificada é verdadeira. Nem sempre usamos um par separado de comandos **begin** e **end,** pois não há ambigüidade quando eles são omitidos. (Observe que algumas linguagens de programação não permitem esta omissão.)

Usando Procedimentos em Outros Procedimentos

Podemos usar um procedimento dentro de outro procedimento (ou dentro dele próprio em um programa recursivo) simplesmente escrevendo o nome deste procedimento seguido pelas entradas deste procedimento. Por exemplo,

```
max(L)
```

executará o procedimento *max* com a lista de entrada *L*. Depois de todos os passos deste procedimento terem sido executados, a execução continua com o próximo comando neste procedimento.

Exercícios

1. Qual é a diferença entre os seguintes blocos de dois comandos de atribuição de valores?

$a := b$
$b := c$

e

$b := c$
$a := b$

2. Dê um procedimento usando comandos de atribuição de valores para trocar os valores das variáveis x e y. Qual o número mínimo de comandos de atribuição de valores necessários para fazer isso?

3. Mostre como um laço da forma

for ***i*** := *valor inicial* **to** *valor final*
comando

pode ser escrito usando a construção "while".

Sugestões de Leitura

Entre as fontes disponíveis para aprender mais sobre os tópicos abordados neste livro estão materiais impressos e endereços relevantes na Web. Os materiais eletrônicos estão disponíveis em inglês, no site do livro, em www.mhhe.com/rosen. As fontes impressas estão descritas nesta seção de sugestões de leitura. As leituras estão relacionadas por capítulo e ligadas a tópicos particulares de interesse. Algumas referências gerais também merecem menção especial. Um livro que você pode achar particularmente útil é o Handbook of Discrete e Combinatorial Mathematics, de Rosen [Ro00], livro de referência abrangente. Uma cobertura mais profunda de muitos tópicos na ciência da computação, incluindo aqueles discutidos neste livro, pode ser encontrada em Gruska [Gr97]. Informação bibliográfica sobre muitos dos matemáticos e cientistas da computação mencionados neste livro pode ser encontrada em Gillispie [Gi70].

Para encontrar websites pertinentes, consulte os links situados no link Web Resources Guide, no site do livro, em www.mhhe.com/rosen. Tudo disponível em inglês.

CAPÍTULO 1

Uma maneira divertida de estudar lógica é ler o livro de Lewis Carroll [Ca78]. Referências gerais para lógica incluem M. Huth e M. Ryan [HuRy04], Mendelson [Me64], Stoll [St74] e Suppes [Su87]. Um tratamento abrangente da lógica na matemática discreta pode ser encontrado em Gries e Schneider [GrSc93]. Especificações de sistemas são discutidas em Ince [In93]. Os quebra-cabeças de Smullyan sobre cavaleiros e ladrões foram introduzidos em [Sm78]. Ele escreveu muitos livros fascinantes sobre quebra-cabeças lógicos, incluindo [Sm92] e [Sm98]. Prolog é discutido em profundidade em Nilsson e Maluszynski [NiMa95] e em Clocksin e Mellish [ClMe94]. As demonstrações básicas são abordadas em Cupillari [Cu05], Morash [Mo91], Solow [So04], Velleman [Ve94] e Wolf [Wo98]. A ciência e a arte de construir demonstrações são discutidas de maneira agradável em três livros de Pólya: [Po62, 71, 90]. Problemas que implicam ladrilhar um tabuleiro de xadrez usando dominós e polio minós são discutidos em Golomb [Go94] e Martin [Ma91].

CAPÍTULO 2

O texto de Lin e Lin [LiLi81] fornece uma leitura fácil sobre conjuntos e suas aplicações. Os desenvolvimentos axiomáticos da teoria dos conjuntos podem ser encontrados em Halmos [Ha60], Monk [Mo69] e Stoll [St74]. Brualdi [Br04] e Reingold, Nievergelt e Deo [ReNiDe77] contêm introduções a multiconjuntos. Conjuntos fuzzy e suas aplicações a sistemas especialistas e inteligência artificial são tratados em Negoita [Ne85] e Zimmerman [Zi91]. Livros de cálculo, tais como Apostol [Ap67], Spivak [Sp94] e Thomas e Finney [ThFi96], contêm discussões de funções. A melhor fonte impressa de informações sobre seqüências inteiras é Sloan e Plouffe [SlPl95]. Stanat e McAllister [StMc77] têm uma seção completa sobre contagem. Discussões dos fundamentos matemáticos necessários na ciência da computação podem ser encontradas em Arbib, Kfoury e Moll [ArKfMo80], Bobrow e Arbib [BoAr74], Beckman [Be80] e Tremblay e Manohar [TrMa75].

CAPÍTULO 3

Os artigos de Knuth [Kn77] e Wirth [Wi84] são introduções acessíveis ao assunto dos algoritmos. Material extensivo sobre estimativas O grande de funções pode ser encontrado em Knuth [Kn97a]. Referências gerais sobre algoritmos e sua complexidade incluem Aho, Hopcroft e Ullman [AhHoUl74]; Baase e Van Gelder [BaGe99]; Cormen, Leierson, Rivest e Stein [CoLeRiSt01]; Gonnet [Go84]; Goodman e Hedetniemi [GoHe77]; Harel [Ha87]; Horowitz e Sahni [HoSa82]; Kreher e Stinson [KrSt98]; a famosa série de livros de Knuth sobre a arte da programação computacional [Kn97a, b, 98]; Kronsjö [Kr87]; Pohl e Shaw [PoSh81]; Purdom e Brown [PuBr85]; Rawlins [Ra92]; Sedgewick [Se03]; Wilf [Wi02] e Wirth [Wi76]. Algoritmos de busca e ordenação e sua complexidade são estudados em detalhe em Knuth [Kn98]. Referências para a teoria dos números incluem Hardy e Wright [HaWr79], LeVeque [Le77], Rosen [Ro05] e Stark [St78]. Aplicações da teoria dos números à criptografia são discutidas em Denning [De82]; Menezes, van Oorschot e Vanstone [MeOoVa97]; Rosen [Ro05]; Seberry e Pieprzyk [SePi89]; Sinkov [Si66] e Stinson [St05]. O sistema de chave pública RSA foi introduzido em Rivest, Shamir e Adleman [RiShAd78]. Uma agradável descrição da história da criptografia pode ser encontrada em [Si99]. Algoritmos para aritmética computacional são discutidos em Knuth [Kn97b] e Pohl e Shaw [PoSh81]. Matrizes e suas operações são discutidas em todos os livros de álgebra linear, tais como Curtis [Cu84] e Strang [St88].

CAPÍTULO 4

Uma introdução acessível à indução matemática pode ser encontrada na tradução para o inglês (do original em russo) de Sominskii [So61]. Os livros que contêm tratamentos completos de indução matemática e definições recursivas incluem Liu [Li85], Sahni [Sa85], Stanat e McAllister [StMc77] e Tremblay e Manohar [TrMa75]. A função de Ackermann, introduzida em 1928 por W. Ackermann, aparece na teoria das funções recursivas (veja Beckman [Be80] e McNaughton [Mc82], por exemplo) e na análise da complexidade de certos conjuntos de algoritmos teóricos (veja Tarjan [Ta83]). A recursão é estudada em Roberts [Ro86], Rohl [Ro84] e Wand [Wa80]. Discussões sobre a correção de programas e da maquinaria lógica usada para demonstrar que programas estão corretos podem ser encontradas em Alagic e Arbib [AlAr78], Anderson [An79], Backhouse [Ba86], Sahni [Sa85] e Stanat e McAllister [StMc77].

CAPÍTULO 5

Referências gerais para técnicas de contagem e suas aplicações incluem Anderson [An74]; Berman e Fryer [BeFr72]; Bogart [Bo00]; Bona [Bo07]; Bose e Manvel [BoMa86]; Brualdi [Br04]; Cohen [Co78]; Grimaldi [Gr03]; Liu [Li68]; Pólya, Tarjan e Woods [PoTaWo83]; Riordan [Ri58]; Roberts eTesman [RoTe03]; Tucker [Tu02] e Williamson [Wi85]. Vilenkin [Vi71] contém uma seleção de problemas de combinatória e suas soluções. Uma seleção de problemas de combinatória mais difíceis pode ser encontrada em Lovász [Lo79]. Informação sobre endereços de protocolos da internet e datagramas pode ser encontrada em Comer [Co05]. Aplicações do princípio da casa dos pombos podem ser encontradas em Brualdi [Br04], Liu [Li85] e Roberts e Tesman [RoTe03]. Uma ampla seleção de identidades combinatórias pode ser encontrada em Riordan [Ri68] e em Benjamin e Quinn [BeQu03]. Algoritmos combinatórios, incluindo algoritmos para gerar permutações e combinações, são descritos por Even [Ev73]; Lehmer [Le64] e Reingold, Nievergelt e Deo [ReNiDe77].

CAPÍTULO 6

Referências úteis para a teoria da probabilidade discreta incluem Feller [Fe68], Nabin [Na00] e Ross [Ro05a]. Ross [Ro02], que se concentra na aplicação da teoria da probabilidade à ciência da computação, fornece exemplos de casos médios na análise de complexidade e abrange os métodos probabilísticos. Aho e Ullman [AhUl95] incluem uma discussão de vários aspectos da teoria da probabilidade importantes na ciência da computação, incluindo aplicações da probabilidade em programação. O método probabilístico é discutido em um capítulo de Aigner e Ziegler [AiZi04], uma monografia dedicada a demonstrações inteligentes, intuitivas e brilhantes, ou seja, demonstrações que Paul Erdös descreveu como vindas de "O Livro". Uma abordagem abrangente do método probabilístico pode ser encontrada em Alon e Spencer [AlSp00]. O Teorema de Bayes é discutido em [PaPi01]. Material adicional sobre filtros de spam pode ser encontrado em [Zd05].

CAPÍTULO 7

Muitos modelos diferentes que usam relações de recorrência podem ser encontrados em Roberts e Tesman [RoTe03] e Tucker [Tu02]. Um tratamento exaustivo das relações de recorrência homogêneas lineares com coeficientes constantes e relações de recorrência não homogêneas relacionadas pode ser encontrado em Brualdi [Br04], Liu [Li68] e Mattson [Ma93]. Algoritmos de divisão e resolução e sua complexidade são discutidos em Roberts e Tesman [RoTe03] e Stanat e McAllister [StMc77]. Descrições de multiplicação rápida de inteiros e matrizes podem ser encontradas em Aho, Hopcroft, e Ullman [AhHoUl74] e Knuth [Kn97b]. Uma introdução excelente às funções geradoras pode ser encontrada em Pólya, Tarjan e Woods [PoTaWo83]. As funções geradoras são estudadas em detalhe em Brualdi [Br04]; Cohen [Co78]; Graham, Knuth e Patashnik [GrKnPa94]; Grimaldi [Gr03] e Roberts e Tesman [RoTe03]. Aplicações adicionais do princípio de inclusão–exclusão podem ser encontradas em Liu [Li85, Li68], Roberts e Tesman [RoTe03] e Ryser [Ry63].

CAPÍTULO 8

Referências gerais para relações, incluindo tratamentos das relações de equivalência e de ordens parciais, incluem Bobrow e Arbib [BoAr74], Grimaldi [Gr03], Sanhi [Sa85] e Tremblay e Manohar [TrMa75]. Discussões de modelos relacionais para banco de dados são dadas em Date [Da82] e Aho e Ullman [AhUl95]. Os artigos originais de Roy e Warshall para encontrar fechos transitivos podem ser encontrados em [Ro59] e [Wa62], respectivamente. Grafos orientados são estudados em Chartrand e Lesniak [ChLe96], Gross e Yellen [GrYe05], Robinson e Foulds [RoFo80], Roberts e Tesman [RoTe03] e Tucker [Tu85]. A aplicação de reticulados ao fluxo de informação é tratada em Denning [De82].

CAPÍTULO 9

Referências gerais para a teoria dos grafos incluem Behzad e Chartrand [BeCh71], Chartrand e Lesniak [ChLe96], Chartrand e Zhang [ChZh04], Bondy e Murty [BoMu76], Chartrand e Oellermann [ChOe93], Graver e Watkins [GrWa77], Gross e Yellen [GrYe03, 05], Harary [Ha69], Ore [Or63], Roberts e Tesman [RoTe03], Tucker [Tu02],

Wilson [Wi85] e Wilson e Watkins [WiWa90]. Uma ampla variedade de aplicações da teoria dos grafos pode ser encontrada em Chartrand [Ch77], Deo [De74], Foulds [Fo92], Roberts e Tesman [RoTe03], Roberts [Ro76], Wilson e Beineke [WiBe79] e McHugh [Mc90]. Aplicações que envolvem grafos grandes, incluindo o grafo da Web, são discutidas em Hayes [Ha00a, 00b]. Uma descrição abrangente dos algoritmos na teoria dos grafos pode ser encontrada em Gibbons [Gi85]. Outras referências para algoritmos na teoria dos grafos incluem Buckley e Harary [BuHa90]; Chartrand e Oellermann [ChOe93]; Chachra, Ghare e Moore [ChGhMo79]; Even [Ev73, 79]; Hu [Hu82] e Reingold, Nievergelt e Deo [ReNiDe77]. Uma tradução do artigo original de Euler sobre o problema das pontes de Königsberg pode ser encontrada em [Eu53]. O algoritmo de Dijkstra é estudado em Gibbons [Gi85];

Liu [Li85] e Reingold, Nievergelt e Deo [ReNiDe77]. O artigo original de Dijkstra pode ser encontrado em [Di59]. Uma demonstração do Teorema de Kuratowski pode ser encontrada em Harary [Ha69] e Liu [Li68]. Números de cruzamento e espessura de grafos são estudados em Chartrand e Lesniak [ChLe96]. Referências para a coloração de grafos e o teorema das quatro cores são incluídas em Barnette [Ba83] e Saaty e Kainen [SaKa86]. O triunfo original do teorema das quatro cores é relatado em Appel e Haken [ApHa76]. Aplicações da coloração de grafos são descritas por Roberts e Tesman [RoTe03]. A história da teoria dos grafos é tratada em Biggs, Lloyd e Wilson [BiLlWi86]. As redes de interconexão para processamento paralelo são discutidas em Akl [Ak89] e Siegel e Hsu [SiHs88].

CAPÍTULO 10

As árvores são estudadas em Deo [De74], Grimaldi [Gr03], Knuth [Kn97a], Roberts e Tesman [RoTe03] e Tucker [Tu85]. O uso de árvores em ciência da computação é descrito por Gotlieb e Gotlieb [GoGo78], Horowitz e Sahni [HoSa82] e Knuth [Kn97a, 98]. Roberts e Tesman [RoTe03] tratam das aplicações de árvores a muitas áreas diferentes. Códigos de prefixo e codificação de Huffman são abrangidos por Hamming [Ha80]. *Backtracking* é uma técnica antiga; seu uso para resolver quebra-cabeças de labirintos pode ser encontrado no livro de 1891 de Lucas [Lu91]. Uma discussão abrangente de como resolver problemas usando o *backtracking* pode ser encontrada em Reingold, Nievergelt e Deo [ReNiDe77]. Gibbons [Gi85] e Reingold, Nievergelt e Deo [ReNiDe77] contêm discussões de algoritmos para construir árvores geradoras e árvores geradoras mínimas. Os antecedentes e a história dos algoritmos para encontrar árvores geradoras mínimas são tratados em Graham e Hell [GrHe85]. Prim e Kruskal descreveram seus algoritmos para encontrar árvores geradoras mínimas em [Pr57] e [Kr56], respectivamente. O algoritmo de Sollin é um exemplo de um algoritmo adequado para processamento paralelo; embora Sollin nunca tenha publicado uma descrição dele, seu algoritmo foi descrito por Even [Ev73] e Goodman e Hedetniemi [GoHe77].

CAPÍTULO 11

A álgebra booleana é estudada em Hohn [Ho66], Kohavi [Ko86] e Tremblay e Manohar [TrMa75]. Aplicações da álgebra booleana aos circuitos lógicos e circuitos de chaves são descritas por Hayes [Ha93], Hohn [Ho66], Katz [Ka93] e Kohavi [Ko86]. Os artigos originais que tratam da minimização das expansões em soma de produtos usando mapas estão em Karnaugh [Ka53] e Veitch [Ve52]. O método de Quine-McCluskey foi introduzido em McCluskey [Mc56] e Quine [Qu52, 55]. Funções *threshold* são abordadas em Kohavi [Ko86].

CAPÍTULO 12

Referências gerais para gramáticas formais, teoria de autômatos e teoria da computação incluem Davis, Sigal e Weyuker [DaSiWe94]; Denning, Dennis e Qualitz [DeDeQu81]; Hopcroft, Motwani e Ullman [HoMoUl01]; Hopkin e Moss [HoMo76]; Lewis e Papadimitriou [LePa97]e McNaughton [Mc82] e Sipser [Si06]. Máquinas de Mealy e máquinas de Moore foram introduzidas originalmente em Mealy [Me55] e Moore [Mo56]. A demonstração original do teorema de Kleene pode ser encontrada em [Kl56]. Modelos poderosos da computação, incluindo o autômato de pilha e as máquinas de Turing, são discutidos em Brookshear [Br89], Hennie [He77], Hopcroft e Ullman [HoUl79], Hopkin e Moss [HoMo76], Martin [Ma03], Sipser [Si06] e Wood [Wo87]. Barwise e Etchemendy [BaEt93] é uma excelente introdução às máquinas de Turing. Artigos interessantes sobre história e aplicação das máquinas de Turing e máquinas relacionadas podem ser encontrados em Herken [He88]. Máquinas *busy beaver* foram introduzinas inicialmente por Rado em [Ra62] e informações sobre elas podem ser encontradas em Dewdney [De84, 93], no artigo de Brady em Herken [He88] e em Wood [Wo87].

APÊNDICES

Uma discussão dos axiomas para os números reais e para os inteiros pode ser encontrada em Morash [Mo91]. Tratamentos detalhados das funções exponenciais e logarítmicas podem ser encontrados em livros de cálculo, tais como Apostol [Ap67], Spivak [Sp94] e Thomas e Finney [ThFi96]. O pseudocódigo descrito no Apêndice 3 se parece com o Pascal. Pohl e Shaw [PoSh81] usam um tipo de linguagem parecida para descrever algoritmos. Wirth [Wi76] descreve como usar algoritmos e estruturas de dados para construir programas usando Pascal. Além

disso, Rohl [Ro84] mostra como implementar programas recursivos usando Pascal.

REFERÊNCIAS

[AhHoUl74] A. V. Aho, J. E. Hopcroft e J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Reading, MA: Addison-Wesley, 1974.

[AhUl95] Alfred V. Aho e Jeffrey D. Ullman. *Foundations of Computer Science, C Edition.* Nova York: Computer Science Press, 1995.

[AiZi04] Martin Aigner e Günter M. Ziegler. *Proofs from THE BOOK.* 3. ed. Berlim: Springer, 2004.

[Ak89] S. G. Akl. *The Design and Analysis of Parallel Algorithms.* Englewood Cliffs, NJ: Prentice Hall, 1989.

[AlAr78] S. Alagic e M. A. Arbib. *The Design of Well-Structured and Correct Programs.* Nova York: Springer-Verlag, 1978.

[AlSp00] Noga Alon e Joel H. Spencer. *The Probabilistic Method.* 2. ed. Nova York:Wiley, 2000.

[An74] I. Anderson. *A First Course in Combinatorial Mathematics.* Oxford, Inglaterra: Clarendon, 1974.

[An79] R. B. Anderson. *Proving Programs Correct.* Nova York: Wiley, 1979.

[Ap67] T. M. Apostol. *Calculus.* 2. ed. Nova York: Wiley, 1967. v. I.

[ApHa76] K. Appel e W. Haken. "Every Planar Map Is 4-colorable", *Bulletin of the AMS,* 82, p. 711–712, 1976.

[ArKfMo80] M. A. Arbib, A. J. Kfoury e R. N. Moll. *A Basis for Theoretical Computer Science.* Nova York: Springer-Verlag, 1980.

[BaGe99] S. Baase e A.Van Gelder. *Computer Algorithms: Introduction to Design and Analysis.* 3. ed. Reading, MA: Addison-Wesley, 1999.

[Ba86] R. C. Backhouse. *Program Construction and Verification.* Englewood Cliffs, NJ: Prentice-Hall, 1986.

[Ba83] D. Barnette. *Map Coloring, Polyhedra, and the Four-Color Problem.* Washington, DC: Mathematical Association of America, 1983.

[BaEt93] Jon Barwise e John Etchemendy. *Turing's World 3.0 for the Macintosh.* Stanford, CA: CSLI Publications, 1993.

[Be80] F. S. Beckman. *Mathematical Foundations of Programming.* Reading, MA: Addison-Wesley, 1980.

[BeCh71] M. Behzad e G. Chartrand. *Introduction to the Theory of Graphs.* Boston: Allyn & Bacon, 1971.

[BeQu03] A. Benjamin e J. J. Quine. *Proofs that Really Count.* Washington, DC: Mathematical Association of America, 2003.

[Be86] J. Bentley. *Programming Pearls.* Addison-Wesley, Reading, MA, 1986.

[BeFr72] G. Berman e K. D. Fryer. *Introduction to Combinatorics.* Nova York: Academic Press, 1972.

[BiLlWi86] N. L. Biggs, E. K. Lloyd e R. J.Wilson. *Graph Theory 1736–1936.* Oxford, Inglaterra: Clarendon, 1986.

[BoAr74] L. S. Bobrow e M. A. Arbib. *Discrete Mathematics.* Filadélfia: Saunders, 1974.

[Bo00] K. P. Bogart. *Introductory Combinatorics.* 3. ed. San Diego: Academic Press, 2000.

[Bo07] M. Bona. *Enumerative Combinatorics.* Nova York: McGraw-Hill, 2007.

[BoMu76] J. A. Bondy e U. S. R. Murty. *Graph Theory with Applications.* Nova York: North-Holland, 1976.

[BoMa86] R. C. Bose e B. Manvel. *Introduction to Combinatorial Theory.* Nova York: Wiley, 1986.

[Br89] J. G. Brookshear. *Theory of Computation.* Redwood City, CA: Benjamin Cummings, 1989.

[Br04] R. A. Brualdi. *Introductory Combinatorics.* 4. ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.

[BuHa90] F. Buckley e F. Harary. *Distance in Graphs.* Redwood City, CA: Addison-Wesley, 1990.

[Ca79] L. Carmony. "Odd Pie Fights", *Mathematics Teacher,* 72, p. 61–64, jan. 1979.

[Ca78] L. Carroll. *Symbolic Logic.* Nova York: Crown, 1978.

[ChGhMo79] V. Chachra, P. M. Ghare e J. M. Moore. *Applications of Graph Theory Algorithms.* Nova York: North-Holland, 1979.

[Ch77] G. Chartrand. *Graphs as Mathematical Models.* Boston: Prindle, Weber & Schmidt, 1977.

[ChLe96] G. Chartrand e L. Lesniak. *Graphs and Digraphs.* 3. ed. Belmont, CA: Wadsworth, 1996.

[ChOe93] G. Chartrand e O. R. Oellermann. *Applied Algorithmic Graph Theory.* Nova York: McGraw-Hill, 1993.

[ChZh04] G. Chartrand e P Zhang. *Introduction to Graph Theory.* Nova York: McGraw-Hill, 2004.

[ClMe94] W. F. Clocksin e C. S. Mellish. *Programming in Prolog.* 4. ed., Nova York: Springer-Verlag, 1994.

[Co78] D. I. A. Cohen. *Basic Techniques of Combinatorial Theory.* Nova York: Wiley, 1978.

[Co05] D. Comer. *Internetworking with TCP/IP: Principles, Protocols, and Architecture.* 5. ed. Englewood Cliffs, NJ: Prentice-Hall, 2005. v. 1.

[CoLeRiSt01] T. H. Cormen, C. E. Leierson, R. L. Rivest e C. Stein. *Introduction to Algorithms.* 2. ed. Cambridge, MA: MIT Press, 2001.

[CrPo01] Richard Crandall e Carl Pomerance. *Prime Numbers: A Computational Perspective.* Nova York: Springer-Verlag, 2001.

[Cu05] Antonella Cupillari. *The Nuts and Bolts of Proofs.* 3. ed. San Diego: Academic Press, 2005.

[Cu84] C. W. Curtis. *Linear Algebra.* Nova York: Springer-Verlag, 1984.

[Da82] C. J. Date. *An Introduction to Database Systems.* 3. ed. Reading, MA: Addison-Wesley, 1982.

[DaSiWe94] M. Davis, R. Sigal e E. J. Weyuker. *Computability, Complexity, and Languages.* 2. ed. San Diego: Academic Press, 1994.

[De82] D. E. R. Denning. *Cryptography and Data Security.* Reading, MA: Addison-Wesley, 1982.

[DeDeQu81] P. J. Denning, J. B. Dennis e J. E. Qualitz. *Machines, Languages, and Computation.* Englewood Cliffs, NJ: Prentice-Hall, 1981.

[De74] N. Deo. *Graph Theory with Applications to Engineering and Computer Science.* Englewood Cliffs, NJ: Prentice-Hall, 1974.

[De02] K. Devlin. *The Millennium Problems: The Seven Greatest Unsolved Mathematical Puzzles of Our Time.* Nova York: Basic Book, 2002.

[De84] A. K. Dewdney. "Computer Recreations", *Scientific American,* 251, n. 2, p. 19–23, ago. 1984; 252, n. 3, p. 14–23, mar. 1985; 251, n. 4 p. 20–30, abr. 1985.

[De93] A. K. Dewdney. *The New Turing Omnibus: Sixty-Six Excursions in Computer Science.* Nova York: W. H. Freeman, 1993.

[Di59] E. Dijkstra. "Two Problems in Connexion with Graphs", *Numerische Mathematik,* 1, p. 269–271, 1959.

[Eu53] L. Euler. "The Koenigsberg Bridges", *Scientific American,* 189, n. 1, p. 66–70, jul. 1953.

[Ev73] S. Even. *Algorithmic Combinatorics.* Nova York: Macmillan, 1973.

[Ev79] S. Even. *Graph Algorithms.* Rockville, MD: Computer Science Press, 1979.

[Fe68] W. Feller. *An Introduction to Probability Theory and Its Applications.* 3. ed. Nova York: Wiley, 1968. v. 1.

[Fo92] L. R. Foulds. *Graph Theory Applications.* Nova York: Springer-Verlag, 1992.

[GaJo79] Michael R. Garey e David S. Johnson. *Computers and Intractability: A Guide to NP-Completeness.* Nova York: Freeman, 1979.

[Gi85] A. Gibbons. *Algorithmic Graph Theory.* Cambridge, Inglaterra: Cambridge University Press, 1985.

[Gi70] C. C. Gillispie (Ed.). *Dictionary of Scientific Biography.* Nova York: Scribner's, 1970.

[Go94] S. W. Golomb. *Polyominoes.* Princeton, NJ: Princeton University Press, 1994.

[Go84] G. H. Gonnet. *Handbook of Algorithms and Data Structures.* Londres: Addison-Wesley, 1984.

[GoHe77] S. E. Goodman e S. T. Hedetniemi. *Introduction to the Design and Analysis of Algorithms.* Nova York: McGraw-Hill, 1977.

[GoGo78] C. C. Gotlieb e L. R. Gotlieb. *Data Types and Structures.* Englewood Cliffs, NJ: Prentice-Hall, 1978.

[GrHe85] R. L. Graham e P. Hell. "On the History of the Minimum Spanning Tree Problem", *Annals of the History of Computing,* 7, p. 43–57, 1985.

[GrKnPa94] R. L. Graham, D. E. Knuth e O. Patashnik. *Concrete Mathematics.* 2. ed. Reading, MA: Addison-Wesley, 1994.

[GrRoSp90] Ronald L. Graham, Bruce L. Rothschild e Joel H. Spencer. *Ramsey Theory.* 2. ed. Nova York: Wiley, 1990.

[GrWa77] J. E. Graver e M. E. Watkins. *Combinatorics with Emphasis on the Theory of Graphs.* Nova York: Springer-Verlag, 1977.

[GrSc93] D. Gries e F. B. Schneider. *A Logical Approach to Discrete Math.* Nova York: Springer-Verlag, 1993.

[Gr03] R. P. Grimaldi. *Discrete and Combinatorial Mathematics.* 5. ed. Reading, MA: Addison-Wesley, 2003.

[GrYe05] J. L. Gross e J. Yellen. *Graph Theory and Its Applications.* 2. ed. Boca Raton, FL: CRC Press, 2005.

[GrYe03] J. L. Gross e J. Yellen. *Handbook of Graph Theory.* Boca Raton, FL: CRC Press, 2003.

[Gr90] Jerrold W. Grossman. *Discrete Mathematics: An Introduction to Concepts, Methods, and Applications.* Nova York: Macmillan, 1990.

[Gr97] J. Gruska. *Foundations of Computing.* Londres: International Thomsen Computer Press, 1997.

[Ha60] P. R. Halmos. *Naive Set Theory.* Nova York: D. Van Nostrand, 1960.

[Ha80] R. W. Hamming. *Coding and Information Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1980.

[Ha69] F. Harary. *Graph Theory.* Reading, MA: Addison-Wesley, 1969.

[HaWr79] G. H. Hardy e E. M. Wright. *An Introduction to the Theory of Numbers.* 5. ed. Oxford, Inglaterra: Oxford University Press, 1979.

[Ha87] D. Harel. *Algorithmics, The Spirit of Computing.* Reading, MA: Addison-Wesley, 1987.

[Ha00a] Brian Hayes. "Graph-Theory in Practice, Part I", *American Scientist,* 88, n. 1, p. 9–13, 2000.

[Ha00b] Brian Hayes. "Graph-Theory in Practice, Part II", *American Scientist,* 88, n. 2, p. 104–109, 2000.

[Ha93] John P. Hayes. *Introduction to Digital Logic Design.* Reading, MA: Addison-Wesley, 1993.

[He77] F. Hennie. *Introduction to Computability.* Reading, MA: Addison-Wesley, 1977.

[He88] R. Herken. *The Universal Turing Machine, A Half-Century Survey.* Nova York: Oxford University Press, 1988.

[Ho76] C. Ho. "Decomposition of a Polygon into Triangles", *Mathematical Gazette,* 60, p. 132–134, 1976.

[Ho99] D. Hofstadter. *Gödel, Escher, Bach: An Internal Golden Braid.* Nova York: Basic Books, 1999.

[Ho66] F. E. Hohn. *Applied Boolean Algebra.* 2. ed. Nova York: Macmillan, 1966.

[HoMoUl01] J. E. Hopcroft e J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* 2. ed. Reading, MA: Addison-Wesley, 1979.

[HoMo76] D. Hopkin e B. Moss. *Automata.* Nova York: Elsevier, North-Holland, 1976.

[HoSa82] E. Horowitz e S. Sahni. *Fundamentals of Computer Algorithms.* Rockville, MD: Computer Science Press, 1982.

[Hu82] T. C. Hu. *Combinatorial Algorithms.* Reading, MA: Addison-Wesley, 1982.

[HuRy04] M. Huth e M. Ryan. *Logic in Computer Science.* 2. ed. Cambridge, Inglaterra: Cambridge University Press, 2004.

[In93] D. C. Ince. *An Introduction to Discrete Mathematics, Formal System Specification, and Z.* 2. ed. Nova York: Oxford, 1993.

[KaMo64] D. Kalish e R. Montague. *Logic: Techniques of Formal Reasoning.* Nova York: Harcourt, Brace, Jovanovich, 1964.

[Ka53] M. Karnaugh. "The Map Method for Synthesis of Combinatorial Logic Circuits", *Transactions of the AIEE,* parte I, 72, p. 593–599, 1953.

[Ka93] Randy H. Katz. *Contemporary Logic Design.* Reading, MA: Addison-Wesley, 1993.

[Kl56] S. C. Kleene. "Representation of Events by Nerve Nets", em *Automata Studies,* Princeton, NJ: Princeton University Press, 1956. p. 3–42.

[Kn77] D. E. Knuth. "Algorithms", *Scientific American,* 236, n. 4, p. 63–80, abr. 1977.

[Kn97a] D. E. Knuth. *The Art of Computer Programming: Fundamental Algorithms.* 3. ed. Reading, MA: Addison-Wesley, 1997a. v. I.

[Kn97b]D. E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms.* 3. ed. Reading, MA: Addison-Wesley, 1997b. v. II.

[Kn98] D. E. Knuth. *The Art of Computer Programming: Sorting e Searching.* 2. ed. Reading, MA: Addison-Wesley, 1998. v. III.

[Ko86] Z. Kohavi. *Switching and Finite Automata Theory.* 2. ed. Nova York: McGraw-Hill, 1986.

[KrSt98] Donald H. Kreher e Douglas R. Stinson. *Combinatorial Algorithms: Generation, Enumeration, and Search.* Boca Raton, FL: CRC Press, 1998.

[Kr87] L. Kronsjö. *Algorithms: Their Complexity and Efficiency.* 2. ed. Nova York: Wiley, 1987.

[Kr56] J. B. Kruskal. "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem", *Proceedings of the AMS,* 1, p. 48–50, 1956.

[Le64] D. H. Lehmer. "The Machine Tools of Combinatorics", em E. F. Beckenbach (Ed.). *Applied Combinatorial Mathematics.* Nova York: Wiley, 1964.

[Le77] W. J. LeVeque. *Fundamentals of Number Theory.* Reading, MA: Addison-Wesley, 1977.

[LePa97] H. R. Lewis e C. H. Papadimitriou. *Elements of the Theory of Computation.* 2. ed. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[LiLi81] Y. Lin e S. Y. T. Lin. *Set Theory with Applications.* 2. ed. Tampa, FL: Mariner, 1981.

[Li68] C. L. Liu. *Introduction to Combinatorial Mathematics.* Nova York: McGraw-Hill, 1968.

[Li85] C. L. Liu. *Elements of Discrete Mathematics.* 2. ed. Nova York: McGraw-Hill, 1985.

[Lo79] L. Lovász. *Combinatorial Problems and Exercises.* Amsterdam: North-Holland, 1979.

[Lu91] E. Lucas. *Récreations Mathématiques.* Paris: Gauthier-Villars, 1891.

[Ma91] G. E. Martin. *Polyominoes: A Guide to Puzzles and Problems in Tiling.* Washington, DC: Mathematical Association of America, 1991.

[Ma03] J. C. Martin. *Introduction to Languages and the Theory of Computation.* 3. ed. Nova York: McGraw-Hill, 2003.

[Ma93] H. F. Mattson, Jr. *Discrete Mathematics with Applications.* Nova York: Wiley, 1993.

[Mc56] E. J. McCluskey, Jr. "Minimization of Boolean Functions", *Bell System Technical Journal,* 35, p. 1417–1444, 1956.

[Mc90] J. A. McHugh. *Algorithmic Graph Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1990.

[Mc82] R. McNaughton. *Elementary Computability, Formal Languages, and Automata.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[Me55] G. H. Mealy. "A Method for Synthesizing Sequential Circuits", *Bell System Technical Journal,* 34, p. 1045–1079, 1955.

[Me64] E. Mendelson. *Introduction to Mathematical Logic.* Nova York: Van Nostrand Reinhold, 1964.

[MeOoVa97] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone. *Handbook of Applied Cryptography.* Boca Raton, FL: CRC Press, 1997.

[MiRo91] J. G. Michaels e K. H. Rosen. *Applications of Discrete Mathematics.* Nova York: McGraw-Hill, 1991.

[Mo69] J. R. Monk. *Introduction to Set Theory.* Nova York: McGraw-Hill, 1969.

[Mo91] R. P. Morash. *Bridge to Abstract Mathematics.* Nova York: McGraw-Hill, 1991.

[Mo56] E. F. Moore. "Gedanken-Experiments on Sequential Machines", em *Automata Studies,* Princeton, NJ: Princeton University Press, 1956. p. 129–153.

[Na00] Paul J. Nabin. *Duelling Idiots and Other Probability Puzzlers.* Princeton, NJ: Princeton University Press, 2000.

[Ne85] C. V. Negoita. *Expert Systems and Fuzzy Systems.* Menlo Park, CA: Benjamin Cummings, 1985.

[NiMa95] Ulf Nilsson e Jan Maluszynski. *Logic, Programming, and Prolog.* 2. ed. Chichester, Inglaterra: Wiley, 1995.

[Or00] J. O'Rourke. *Computational Geometry in C.* Nova York: Cambridge University Press, 2000.

[Or63] O. Ore. *Graphs and Their Uses.* Washington, DC: Mathematical Association of America, 1963.

[PaPi01] A. Papoulis e S. U. Pillai. *Probability, Random Variables, and Stochastic Processes.* Nova York: McGraw-Hill, 2001.

[Pe87] A. Pelc. "Solution of Ulam's Problem on Searching with a Lie", *Journal of Combinatorial Theory,* Series A, 44, p. 129–140, 1987.

[PoSh81] I. Pohl e A. Shaw. *The Nature of Computation: An Introduction to Computer Science.* Rockville, MD: Computer Science Press, 1981.

[Po62] George Pólya. *Mathematical Discovery.* Nova York: Wiley, 1962. v. 1 e 2.

[Po71] George Pólya. *How to Solve It.* Princeton, NJ: Princeton University Press, 1971.

[Po90] George Pólya. *Mathematics and Plausible Reasoning.* Princeton, NJ: Princeton University Press, 1990.

[PoTaWo83] G. Pólya, R. E. Tarjan e D. R. Woods. *Notes on Introductory Combinatorics.* Boston: Birkhäuser, 1983.

[Pr57] R. C. Prim. "Shortest Connection Networks and Some Generalizations", *Bell System Technical Journal,* 36, p. 1389–1401, 1957.

[PuBr85] P. W. Purdom, Jr. e C. A. Brown. *The Analysis of Algorithms*. Nova York: Holt, Rinehart & Winston, 1985.

[Qu52] W. V. Quine. "The Problem of SimplifyingTruth Functions", *American Mathematical Monthly,* 59, p. 521–531, 1952.

[Qu55] W. V. Quine. "A Way to Simplify Truth Functions", *American Mathematical Monthly,* 62, p. 627–631, 1955.

[Ra62] T. Rado. "On Non-Computable Functions", *Bell System Technical Journal*, p. 877–884, maio 1962.

[Ra92] Gregory J. E. Rawlins. *Compared to What? An Introduction to the Analysis of Algorithms*. Nova York: Computer Science Press, 1992.

[ReNiDe77] E. M. Reingold, J. Nievergelt e N. Deo. *Combinatorial Algorithms: Theory and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[Ri58] J. Riordan. *An Introduction to Combinatorial Analysis*. Nova York: Wiley, 1958.

[Ri68] J. Riordan. *Combinatorial Identities*. Nova York: Wiley, 1968.

[RiShAd78] R. Rivest, A. Shamir e L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the Association for Computing Machinery,* 31, n. 2, p. 120–128, 1978.

[Ro86] E. S. Roberts. *Thinking Recursively*. Nova York: Wiley, 1986.

[Ro76] F. S. Roberts. *Discrete Mathematics Models*. Englewood Cliffs, NJ: Prentice-Hall, 1976.

[RoTe03] F. S. Roberts e B. Tesman. *Applied Combinatorics*. 2. ed. Englewood Cliffs, NJ: Prentice-Hall, 2003.

[RoFo80] D. F. Robinson e L. R. Foulds. *Digraphs: Theory and Techniques*. Nova York: Gordon and Breach, 1980.

[Ro84] J. S. Rohl. *Recursion via Pascal*. Cambridge, Inglaterra: Cambridge University Press, 1984.

[Ro05] K. H. Rosen. *Elementary Number Theory and Its Applications*. 5. ed. Reading, MA: Addison-Wesley, 2005.

[Ro00] K. H. Rosen. *Handbook of Discrete and Combinatorial Mathematics*. Boca Raton, FL: CRC Press, 2000.

[Ro05a] Sheldon M. Ross. *A First Course in Probability Theory*. 7. ed. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[Ro02] Sheldon M. Ross. *Probability Models for Computer Science*. Harcourt/Academic Press, 2002.

[Ro59] B. Roy. "Transitivité et Connexité", *C.R. Acad. Sci. Paris,* 249, p. 216, 1959.

[Ry63] H. Ryser. *Combinatorial Mathematics*. Washington, DC: Mathematical Association of America, 1963.

[SaKa86] T. L. Saaty e P. C. Kainen. *The Four-Color Problem: Assaults and Conquest*. Nova York: Dover, 1986.

[Sa85] S. Sahni. *Concepts in Discrete Mathematics*. Minneapolis: Camelot, 1985.

[SePi89] J. Seberry e J. Pieprzyk. *Cryptography: An Introduction to Computer Security*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[Se03] R. Sedgewick. *Algorithms in Java*. 3. ed. Reading, MA: Addison-Wesley, 2003.

[SiHs88] H. J. Siegel e W. T. Hsu. "Interconnection Networks", em *Computer Architectures,*V. M. Milutinovic (Ed.). Nova York: North-Holland, 1988, p. 225–264.

[Si99] S. Singh. *The Code Book*. Nova York: Doubleday, 1999.

[Si66] A. Sinkov. *Elementary Cryptanalysis*. Washington, DC: Mathematical Association of America, 1966.

[Si06] M. Sipser. *Introduction to the Theory of Computation*. Boston: Course Technology, 2006.

[SlPl95] N. J. A. Sloane e S. Plouffe. *The Encyclopedia of Integer Sequences*. Nova York: Academic Press, 1995.

[Sm78] Raymond Smullyan. *What Is the Name of This Book?: The Riddle of Dracula and Other Logical Puzzles*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

[Sm92] Raymond Smullyan. *Lady or the Tiger? And Other Logic Puzzles Including a Mathematical Novel That Features Godel's Great Discovery*. Nova York: Times Book, 1992.

[Sm98] Raymond Smullyan. *The Riddle of Scheherazade: And Other Amazing Puzzles, Ancient&Modern*. Fort Washington, PA: Harvest Book, 1998.

[So04] Daniel Solow. *How to Read and Do Proofs: An Introduction to Mathematical Thought Processes*. Nova York: Wiley, 2001.

[So61] I. S. Sominskii. *Method of Mathematical Induction*. Nova York: Blaisdell, 1961.

[Sp94] M. Spivak. *Calculus*. 3. ed. Wilmington, DE: Publish or Perish, 1994.

[StMc77] D. Stanat e D. F. McAllister. *Discrete Mathematics in Computer Science*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[St78] H. M. Stark. *An Introduction to Number Theory*. Cambridge, MA: MIT Press, 1978.

[St05] Douglas R. Stinson. *Cryptography, Theory and Practice*. 3. ed. Boca Raton, FL: Chapman and Hall/CRC, 2005.

[St94] P. K. Stockmeyer. "Variations on the Four-Post Tower of Hanoi Puzzle", *Congressus Numerantrum,* 102, p. 3–12, 1994.

[St74] R. R. Stoll. *Sets, Logic, and Axiomatic Theories*. 2. ed. São Francisco: W. H. Freeman, 1974.

[St88] G. W. Strang. *Linear Algebra and Its Applications*. 3. ed. San Diego: Harcourt Brace Jovanovich, 1988.

[Su87] P. Suppes. *Introduction to Logic*. Princeton, NJ: D. Van Nostrand, 1987.

[Ta83] R. E. Tarjan. *Data Structures and Network Algorithms*. Filadélfia: Society for Industrial and Applied Mathematics, 1983.

[ThFi96] G. B. Thomas e R. L. Finney. *Calculus and Analytic Geometry*. 9. ed. Reading, MA: Addison-Wesley, 1996.

[TrMa75] J. P. Tremblay e R. P. Manohar. *Discrete Mathematical Structures with Applications to Computer Science.* Nova York: McGraw-Hill, 1975.

[Tu02] Alan Tucker. *Applied Combinatorics.* 4. ed. Nova York: Wiley, 2002.

[Ve52] E. W. Veitch. "A Chart Method for Simplifying Truth Functions", *Proceedings of the ACM*, p. 127–133, 1952.

[Ve94] David Velleman. *How to Prove It: A Structured Approach.* Nova York: Cambridge University Press, 1994.

[Vi71] N. Y. Vilenkin. *Combinatorics.* Nova York: Academic Press, 1971.

[Wa80] M. Wand. *Induction, Recursion, and Programming.* Nova York: North-Holland, 1980.

[Wa62] S.Warshall. "A Theorem on Boolean Matrices", *Journal of the ACM,* 9, p. 11–12, 1962.

[Wi02] Herbert S. Wilf. *Algorithms and Complexity.* 2. ed. Natick, MA: A. K. Peters, 2002.

[Wi85] S. G. Williamson. *Combinatorics for Computer Science.* Rockville, MD: Computer Science Press, 1985.

[Wi85] R. J. Wilson. *Introduction to Graph Theory.* 3. ed. Longman, Essex, Inglaterra, 1985.

[WiBe79] R. J. Wilson e L. W. Beineke. *Applications of Graph Theory.* Londres: Academic Press, 1979.

[WiWa90] R. J. Wilson e J. J. Watkins. *Graphs, An Introductory Approach.* Nova York: Wiley, 1990.

[Wi76] N. Wirth. *Algorithms + Data Structures = Programs.* Englewood Cliffs, NJ: Prentice-Hall, 1976.

[Wi84] N.Wirth. "Data Structures and Algorithms", *Scientific American,* 251, p. 60–69, set. 1984.

[Wo98] Robert S. Wolf. *Proof, Logic, and Conjecture: The Mathematician's Toolbox.* Nova York: W. H. Freeman, 1998.

[Wo87] D. Wood. *Theory of Computation.* Nova York: Harper & Row, 1987.

[Zd05] J. Zdziarski. *Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification.* São Francisco: No Starch Press, 2005.

[Zi91] H. J. Zimmermann. *Fuzzy Set Theory and Its Applications.* 2. ed. Boston: Kluwer, 1991.

Respostas dos Exercícios de Número Ímpar

Capítulo 1

Seção 1.1

1. a) Sim, V **b)** Sim, F **c)** Sim, V **d)** Sim, F **e)** Não **f)** Não **3. a)** Hoje não é quinta-feira. **b)** Há poluição em São Paulo. **c)** $2 + 1 \neq 3$. **d)** O verão no Rio não é quente nem ensolarado. **5. a)** Não foram descobertos tubarões perto da praia. **b)** Nadar na praia em Nova Jersey é permitido, e foram descobertos tubarões perto da praia. **c)** Nadar na praia em Nova Jersey não é permitido, ou foram descobertos tubarões perto da praia. **d)** Se nadar na praia em Nova Jersey é permitido, então não foram descobertos tubarões perto da praia. **e)** Se tubarões não foram descobertos perto da praia, então nadar na praia em Nova Jersey é permitido. **f)** Se nadar na praia em Nova Jersey não é permitido, então foram descobertos tubarões perto da praia. **g)** Nadar na praia em Nova Jersey é permitido se e somente se não foram descobertos tubarões perto da praia. **h)** Nadar na praia em Nova Jersey não é permitido, e ou nadar na praia em Nova Jersey é permitido ou foram descobertos tubarões perto da praia. (Observe que fomos capazes de incorporar os parênteses pelo uso do primeiro "ou" na segunda metade da sentença.) **7. a)** $p \wedge q$ **b)** $p \wedge \neg q$ **c)** $\neg p \wedge \neg q$ **d)** $p \vee q$ **e)** $p \rightarrow q$ **f)** $(p \vee q) \wedge (p \rightarrow \neg q)$ **g)** $q \leftrightarrow p$ **9. a)** $\neg p$ **b)** $p \wedge \neg q$ **c)** $p \rightarrow q$ **d)** $\neg p \rightarrow \neg q$ **e)** $p \rightarrow q$ **f)** $q \wedge \neg p$ **g)** $q \rightarrow p$ **11. a)** $r \wedge \neg p$ **b)** $\neg p \wedge q \wedge r$ **c)** $r \rightarrow (q \leftrightarrow \neg p)$ **d)** $\neg q \wedge \neg p \wedge r$ **e)** $(q \rightarrow (\neg r \wedge \neg p)) \wedge \neg((\neg r \wedge \neg p) \rightarrow q)$ **f)** $(p \wedge r) \rightarrow \neg q$ **13. a)** Falsa **b)** Verdadeira **c)** Verdadeira **d)** Verdadeira **15. a)** Ou exclusivo: você recebe apenas uma bebida. **b)** Ou inclusivo: senhas longas podem ter qualquer combinação de símbolos. **c)** Ou inclusivo: um estudante com ambos os cursos está ainda mais qualificado. **d)** Ambas as interpretações são possíveis; um viajante poderia querer pagar com uma mistura das duas moedas, ou a loja poderia não permitir isso. **17. a)** Ou inclusivo: é permitido cursar matemática discreta se você fez cálculo ou ciência da computação ou ambos. Ou exclusivo: é permitido cursar matemática discreta se você fez cálculo ou ciência da computação, mas não se você fez ambos. É mais provável que a intenção tenha sido usar o ou inclusivo. **b)** Ou inclusivo: você pode ter o desconto, ou você pode conseguir uma taxa de juros baixa, ou você pode ter tanto o desconto quanto uma taxa de juros baixa. Ou exclusivo: você pode ter o desconto, ou você pode conseguir uma taxa de juros baixa, mas você não pode ter tanto o desconto quanto uma taxa de juros baixa. É mais provável que a intenção tenha sido usar o ou exclusivo. **c)** Ou inclusivo: você pode pedir dois itens da coluna A e nenhum da coluna B, ou três itens da coluna B e nenhum da coluna A, ou cinco itens, incluindo dois da coluna A e três da coluna B. Ou exclusivo: você pode pedir dois itens, da coluna A ou três itens da coluna B, mas não ambos. É quase certo que a intenção tenha sido usar o ou exclusivo. **d)** Ou inclusivo: Mais de dois pés de neve ou sensação térmica do vento abaixo de -100, ou ambos, farão com que a escola seja fechada. Ou exclusivo: Mais de dois pés de neve ou sensação térmica do vento abaixo de -100, mas não ambos, farão com que a escola seja fechada. Com certeza, a intenção foi usar o ou inclusivo. **19. a)** Se o vento soprar do nordeste, então nevará. **b)** Se continuar quente por uma semana, então as macieiras florescerão. **c)** Se o Palmeiras ganhar o campeonato, então ele ganhou do São Paulo. **d)** Se você chegar ao topo do "Pico Long", então você deve ter andado 8 milhas. **e)** Se você for famoso mundialmente, então você conseguirá um mandato como professor. **f)** Se você dirigir por mais de 400 milhas, então você terá de comprar gasolina. **g)** Se a sua garantia for válida, então você deve ter comprado seu aparelho de som em menos de 90 dias. **h)** Se a água não estiver muito fria, então Jan nadará. **21. a)** Você compra um sorvete se e somente se estiver calor lá fora. **b)** Você ganhará na loteria se e somente se você tiver o único bilhete premiado. **c)** Você será promovido se e somente se você tiver contatos. **d)** Sua mente se deteriorará se e somente se você assistir à televisão. **e)** O trem se atrasará se e somente se for um dia que eu pegue o trem. **23. a)** Recíproca: "Eu esquiarei amanhã se nevar hoje." Contrapositiva: "Se eu não esquiar amanhã, então não terá nevado hoje." Inversa: "Se não nevar hoje, então eu não esquiarei amanhã." **b)** Recíproca: "Se eu vier à aula, então haverá uma prova." Contrapositiva: "Se eu não vier à aula, então não haverá uma prova." Inversa: "Se não for haver uma prova, então eu não virei à aula." **c)** Recíproca: "Um inteiro positivo é um primo se ele não tiver divisores além de 1 e dele mesmo." Contrapositiva: "Se um inteiro positivo tiver um divisor além de 1 e dele mesmo, então ele não é primo". Inversa: "Se um inteiro positivo não for primo, então ele tem um divisor além de 1 e dele mesmo."

25. a) 2 **b)** 16 **c)** 64 **d)** 16

27. a)

p	$\neg p$	$p \wedge \neg p$
V	F	F
F	V	F

b)

p	$\neg p$	$p \wedge \neg p$
V	F	V
F	V	V

c)

p	q	$\neg q$	$p \vee \neg q$	$(p \vee \neg q) \rightarrow q$
V	V	F	V	V
V	F	V	V	F
F	V	F	F	V
F	F	V	V	F

d)

p	q	$p \vee q$	$p \wedge q$	$(p \vee q) \rightarrow (p \wedge q)$
V	V	V	V	V
V	F	V	F	F
F	V	V	F	F
F	F	F	F	V

e)

p	q	$p \to q$	$\neg q$	$\neg p$	$\neg q \to \neg p$	$(p \to q) \leftrightarrow (\neg q \to \neg p)$
V	V	V	F	F	V	V
V	F	F	V	F	F	V
F	V	V	F	V	V	V
F	F	V	V	V	V	V

f)

p	q	$p \to q$	$q \to p$	$(p \to q) \to (q \to p)$
V	V	V	V	V
V	F	F	V	V
F	V	V	F	F
F	F	V	V	V

29. Para as partes (a), (b), (c), (d) e (f) temos esta tabela.

p	q	$(p \vee q) \to (p \oplus q)$	$(p \oplus q) \to (p \wedge q)$	$(p \vee q) \oplus (p \wedge q)$	$(p \leftrightarrow q) \oplus (\neg p \leftrightarrow q)$	$(p \oplus q) \to (p \oplus \neg q)$
V	V	F	V	F	V	V
V	F	V	F	V	V	F
F	V	V	F	V	V	F
F	F	V	V	F	V	V

Para a parte (e) temos esta tabela.

p	q	r	$\neg p$	$\neg r$	$p \leftrightarrow q$	$\neg p \leftrightarrow \neg r$	$(p \leftrightarrow q) \oplus (\neg p \leftrightarrow \neg r)$
V	V	V	F	F	V	V	F
V	V	F	F	V	V	F	V
V	F	V	F	F	F	V	V
V	F	F	F	V	F	F	F
F	V	V	V	F	F	F	F
F	V	F	V	V	F	V	V
F	F	V	V	F	V	F	V
F	F	F	V	V	V	V	F

31.

p	q	$p \to \neg q$	$\neg p \leftrightarrow q$	$(p \to q) \vee (\neg p \to q)$	$(p \to q) \wedge (\neg p \to q)$	$(p \leftrightarrow q) \vee (\neg p \leftrightarrow q)$	$(\neg p \leftrightarrow \neg q) \leftrightarrow (p \leftrightarrow q)$
V	V	F	F	V	V	V	V
V	F	V	V	V	F	V	V
F	V	V	V	V	V	V	V
F	F	V	F	V	F	V	V

33.

p	q	r	$p \to (\neg q \vee r)$	$\neg p \to (q \to r)$	$(p \to q) \vee (\neg p \to r)$	$(p \to q) \wedge (\neg p \to r)$	$(p \leftrightarrow q) \vee (\neg q \leftrightarrow r)$	$(\neg p \leftrightarrow \neg q) \leftrightarrow (q \leftrightarrow r)$
V	V	V	V	V	V	V	V	V
V	V	F	F	V	V	V	V	F
V	F	V	V	V	V	F	V	V
V	F	F	V	V	V	F	F	F
F	V	V	V	V	V	V	F	F
F	V	F	V	F	V	F	V	V
F	F	V	V	V	V	V	V	F
F	F	F	V	V	V	F	V	V

35.

p	q	r	s	$p \leftrightarrow q$	$r \leftrightarrow s$	$(p \leftrightarrow q) \leftrightarrow (r \leftrightarrow s)$
V	V	V	V	V	V	V
V	V	V	F	V	F	F
V	V	F	V	V	F	F
V	V	F	F	V	V	V
V	F	V	V	F	V	F
V	F	V	F	F	F	V
V	F	F	V	F	F	V
V	F	F	F	F	V	F
F	V	V	V	F	V	F
F	V	V	F	F	F	V
F	V	F	V	F	F	V
F	V	F	F	F	V	F
F	F	V	V	V	V	V
F	F	V	F	V	F	F
F	F	F	V	V	F	F
F	F	F	F	V	V	V

37. a) A disjunção binária *OR* é 111 1111; a conjunção binária *AND* é 000 0000; a disjunção binária exclusiva *XOR* é 111 1111. **b)** A disjunção binária *OR* é 1111 1010; a conjunção binária *AND* é 1010 0000; a disjunção binária exclusiva *XOR* é 0101 1010. **c)** A disjunção binária *OR* é 10 0111 1001; a conjunção binária *AND* é 00 0100 0000; a disjunção binária exclusiva *XOR* é 10 0011 1001. **d)** A disjunção binária *OR* é 11 1111 1111; a conjunção binária *AND* é 00 0000 0000; a disjunção binária exclusiva *XOR* é 11 1111 1111. **39.** 0,2, 0,6 **41.** 0,8, 0,6 **43. a)** A 99ª afirmação é verdadeira e as outras são falsas. **b)** As afirmações de 1 a 50 são todas verdadeiras e as afirmações de 51 a 100 são todas falsas. **c)** Isso não pode acontecer; é um paradoxo, que mostra que elas não podem ser proposições. **45.** "Se eu lhe perguntasse se a bifurcação à direita leva para as ruínas, você responderia sim?" **47. a)** $q \rightarrow p$ **b)** $q \wedge \neg p$ **c)** $q \rightarrow p$ **d)** $\neg q \rightarrow \neg p$. **49.** Não é consistente. **51.** Consistente. **53.** NOVA **E** JERSEY **E** PRAIAS, (JERSEY **E** PRAIAS) **NÃO** NOVA. **55.** *A* é um cavaleiro e *B* é um bandido. **57.** *A* é um cavaleiro e *B* é um cavaleiro. **59.** *A* é um bandido e *B* é um cavaleiro. **61.** Em ordem decrescente de salário: Fred, Maggie, Janice. **63.** O detetive pode determinar que o mordomo e o cozinheiro estão mentindo, mas não pode determinar se o jardineiro está dizendo a verdade ou se o zelador está dizendo a verdade. **65.** O japonês tem a zebra e o norueguês bebe água.

Seção 1.2

1. As equivalências seguem de se mostrar que os pares apropriados de colunas nesta tabela coincidem.

p	$p \wedge \mathbf{V}$	$p \vee \mathbf{F}$	$p \wedge \mathbf{F}$	$p \vee \mathbf{V}$	$p \vee p$	$p \wedge p$
V	V	V	F	V	V	V
F	F	F	F	V	F	F

3. a)

p	q	$p \vee q$	$q \vee p$
V	V	V	V
V	F	V	V
F	V	V	V
F	F	F	F

b)

p	q	$p \wedge q$	$q \wedge p$
V	V	V	V
T	F	F	F
F	V	F	F
F	F	F	F

5.

p	q	r	$q \vee r$	$p \wedge (q \vee r)$	$p \wedge q$	$p \wedge r$	$(p \wedge q) \vee (p \wedge r)$
V	V	V	V	V	V	V	V
V	V	F	V	V	V	F	V
V	F	V	V	V	F	V	V
V	F	F	F	F	F	F	F
F	V	V	V	F	F	F	F
F	V	F	V	F	F	F	F
F	F	V	V	F	F	F	F
F	F	F	F	F	F	F	F

7. a) Jan não é rica, ou Jan não é feliz. **b)** Carlos não andará de bicicleta amanhã, e Carlos não correrá amanhã. **c)** Mei não anda para ir à escola, e Mei não pega o ônibus para ir à escola. **d)** Ibrahim não é esperto, ou Ibrahim não trabalha muito.

9. a)

p	q	$p \wedge q$	$(p \wedge q) \rightarrow p$
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	V

b)

p	q	$p \vee q$	$p \rightarrow (p \vee q)$
V	V	V	V
V	F	V	V
F	V	V	V
F	F	F	V

c)

p	q	$\neg p$	$p \rightarrow q$	$\neg p \rightarrow (p \rightarrow q)$
V	V	F	V	V
V	F	F	F	V
F	V	V	V	V
F	F	V	V	V

d)

p	q	$p \wedge q$	$p \rightarrow q$	$(p \wedge q) \rightarrow (p \rightarrow q)$
V	V	V	V	V
V	F	F	F	V
F	V	F	V	V
F	F	F	V	V

e)

p	q	$p \to q$	$\neg(p \to q)$	$\neg(p \to q) \to p$
V	V	V	F	V
V	F	F	V	V
F	V	V	F	V
F	F	V	F	V

f)

p	q	$p \to q$	$\neg(p \to q)$	$\neg q$	$\neg(p \to q) \to \neg q$
V	V	V	F	F	V
V	F	F	V	V	V
F	V	V	F	F	V
F	F	V	F	V	V

11. Em cada caso mostraremos que se a hipótese for verdadeira, então a conclusão também é. **a)** Se a hipótese $p \wedge q$ for verdadeira, então, por definição de conjunção, a conclusão p também deve ser verdadeira. **b)** Se a hipótese p for verdadeira, pela definição de disjunção, a conclusão $p \vee q$ também é verdadeira. **c)** Se a hipótese $\neg p$ for verdadeira, isto é, se p é falsa, então a conclusão $p \to q$ é verdadeira. **d)** Se a hipótese $p \wedge q$ for verdadeira, então ambas p e q são verdadeiras, de modo que a conclusão $p \to q$ também é verdadeira. **e)** Se a hipótese $\neg(p \to q)$ for verdadeira, então $p \to q$ é falsa, de modo que a conclusão p é verdadeira (e q é falsa). **f)** Se a hipótese $\neg(p \to q)$ for verdadeira, então $p \to q$ é falsa, de modo que p é verdadeira e q é falsa. Portanto, a conclusão $\neg q$ é verdadeira. **13.** O fato de a quarta coluna da tabela-verdade mostrada ser idêntica à primeira coluna, demonstra a parte **(a)**, e o fato de a sexta coluna ser idêntica à primeira coluna demonstra **(b)**.

p	q	$p \wedge q$	$p \vee (p \wedge q)$	$p \vee q$	$p \wedge (p \vee q)$
V	V	V	V	V	V
V	F	F	V	V	V
F	V	F	F	V	F
F	F	F	F	F	F

15. É uma tautologia. **17.** Cada uma destas é verdadeira exatamente quando p e q têm valores-verdade opostos. **19.** A proposição $\neg p \leftrightarrow q$ é verdadeira quando $\neg p$ e q têm os mesmos valores-verdade, o que significa que p e q têm valores-verdade diferentes. Analogamente, $p \leftrightarrow \neg q$ é verdadeira exatamente nos mesmos casos. Portanto, estas duas expressões são logicamente equivalentes. **21.** A proposição $\neg(p \leftrightarrow q)$ é verdadeira quando $p \leftrightarrow q$ for falsa, o que significa que p e q têm valores-verdade diferentes. Como isso ocorre precisamente quando $\neg p \leftrightarrow q$ for verdadeira, as duas expressões são logicamente equivalentes. **23.** Para $(p \to r) \wedge (q \to r)$ ser falsa, uma das duas proposições condicionais deve ser falsa, o que ocorre exatamente quando r for falsa e, pelo menos, uma entre p e q for verdadeira. Mas estes são precisamente os casos nos quais $p \vee q$ é verdadeira e r é falsa, que é precisamente quando $(p \vee q) \to r$ for falsa. Como as duas proposições são falsas exatamente nas mesmas situações, elas são logicamente equivalentes. **25.** Para $(p \to r) \vee (q \to r)$ ser falsa, as duas proposições condicionais devem ser falsas, o que ocorre exatamente quando r for falsa e ambas p e q forem verdadeiras. Mas este é precisamente o caso em que $p \wedge q$ é verdadeira e r é falsa, que é precisamente quando $(p \wedge q) \to r$ for falsa. Como as duas proposições são falsas exatamente nas mesmas situações, elas são logicamente equivalentes. **27.** Este fato foi observado na Seção 1 quando a bicondicional foi definida inicialmente. Cada uma delas é verdadeira precisamente quando p e q tiverem os mesmos valores-verdade. **29.** A última coluna é toda de Vs.

p	q	r	$p \to q$	$q \to r$	$(p \to q) \wedge (q \to r)$	$p \to r$	$(p \to q) \wedge (q \to r) \to (p \to r)$
V	V	V	V	V	V	V	V
V	V	F	V	F	F	F	V
V	F	V	F	V	F	V	V
V	F	F	F	V	F	F	V
F	V	V	V	V	V	V	V
F	V	F	V	F	F	V	V
F	F	V	V	V	V	V	V
F	F	F	V	V	V	V	V

31. Elas não são logicamente equivalentes, pois quando p, q e r forem todas falsas, $(p \to q) \to r$ é falsa, mas $p \to (q \to r)$ é verdadeira. **33.** São possíveis muitas respostas. Se considerarmos r como verdadeira e p, q e s como falsas, então $(p \to q) \to (r \to s)$ será falsa, mas $(p \to r) \to (q \to s)$ será verdadeira. **35. a)** $p \vee \neg q \vee \neg r$ **b)** $(p \vee q \vee r) \wedge s$ **c)** $(p \wedge \mathbf{V}) \vee (q \wedge \mathbf{F})$ **37.** Se tomarmos duais duas vezes, todo $\vee$ muda para um $\wedge$ e, a seguir, de volta para um $\vee$, todo $\wedge$ muda para um $\vee$ e, a seguir, de volta para um $\wedge$, todo **V** muda para um **F** e, a seguir, de volta para um **V**, todo **F** muda para um **V** e, a seguir, de volta para um **F**. Logo, $(s^*)^* = s$. **39.** Sejam p e q proposições compostas equivalentes que envolvem apenas os operadores $\wedge$, $\vee$ e $\neg$, e **V** e **F**. Observe que $\neg p$ e $\neg q$ também são equivalentes. Use as leis de De Morgan tantas vezes quanto necessário para empurrar as negações tão longe quanto possível para dentro dessas proposições compostas, mudando $\vee$s para $\wedge$s, e vice-versa, e mudando **V**s para **F**s, e vice-versa. Isso mostra que $\neg p$ e $\neg q$ são as mesmas que p^* e q^*, exceto que cada proposição atômica p_i dentro delas é substituída por sua negação. A partir disso, podemos concluir que p^* e q^* são equivalentes, pois $\neg p$ e $\neg q$ são. **41.** $(p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge r) \vee (\neg p \wedge q \wedge r)$ **43.** Dada uma proposição composta p, forme sua tabela-verdade e, a seguir, escreva uma proposição q na forma normal disjuntiva que seja logicamente equivalente a p. Como q envolve apenas $\neg$, $\wedge$ e $\vee$, isso mostra que estes três operadores formam um conjunto funcionalmente completo. **45.** Pelo Exercício 43, dada uma proposição composta p, podemos escrever uma proposição q que é logicamente equivalente a p e envolve apenas $\neg$, $\wedge$ e $\vee$. Pela lei de De Morgan, podemos eliminar todos os $\wedge$s, substituindo cada ocorrência de $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ por $\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n$). **47.** $\neg(p \wedge q)$ é verdadeira quando ou p ou q, ou ambas, forem falsas, e é falsa quando ambas, p e q, forem verdadeiras. Como esta era a definição de $p \mid q$, as duas proposições compostas são logicamente equivalentes. **49.** $\neg(p \vee q)$ é verdadeira quando ambas, p e q, forem falsas, e é falsa caso contrário. Como esta era a definição de $p \downarrow q$, as duas são lo-

gicamente equivalentes. **51.** $((p \downarrow p) \downarrow q) \downarrow ((p \downarrow p) \downarrow q)$ **53.** Isso segue imediatamente da tabela-verdade ou da definição de $p \mid q$. **55.** 16 **57.** Se o banco de dados for aberto, então ou o sistema está em seu estado inicial ou o monitor está colocado em estado de fechamento. **59.** Todas as nove. **61.** Para determinar se c é uma tautologia, aplique um algoritmo para satisfação a $\neg c$. Se o algoritmo disser que $\neg c$ é satisfatória, então respondemos que c não é uma tautologia, e se o algoritmo disser que $\neg c$ não é satisfatória, então respondemos que c é uma tautologia.

Seção 1.3

1. a) V **b)** V **c)** F **3. a)** V **b)** F **c)** F **d)** F **5. a)** Existe um estudante que passa mais do que 5 horas em aula todos os dias da semana. **b)** Todo estudante passa mais do que 5 horas em aula todos os dias da semana. **c)** Existe um estudante que não passa mais do que 5 horas em aula todos os dias da semana. **d)** Nenhum estudante passa mais do que 5 horas em aula todos os dias da semana. **7. a)** Todo comediante é divertido. **b)** Todo mundo é um comediante. **c)** Existe uma pessoa tal que se ela for um comediante, então ela será divertida. **d)** Alguns comediantes são divertidos. **9. a)** $\exists x(P(x) \wedge Q(x))$ **b)** $\exists x(P(x) \wedge \neg Q(x))$ **c)** $\forall x(P(x) \vee Q(x))$ **d)** $\forall x \neg(P(x) \vee Q(x))$ **11. a)** V **b)** V **c)** F **d)** F **e)** V **f)** F **13. a)** Verdadeira **b)** Verdadeira **c)** Verdadeira **d)** Verdadeira **15. a)** Verdadeira **b)** Falsa **c)** Verdadeira **d)** Falsa **17. a)** $P(0) \vee P(1) \vee P(2) \vee P(3) \vee P(4)$ **b)** $P(0) \wedge P(1) \wedge P(2) \wedge P(3) \wedge P(4)$ **c)** $\neg P(0) \vee \neg P(1) \vee \neg P(2) \vee \neg P(3) \vee \neg P(4)$ **d)** $\neg P(0) \wedge \neg P(1) \wedge \neg P(2) \wedge \neg P(3) \wedge \neg P(4)$ **e)** $\neg(P(0) \vee P(1) \vee P(2) \vee P(3) \vee P(4))$ **f)** $\neg(P(0) \wedge P(1) \wedge P(2) \wedge P(3) \wedge P(4))$ **19. a)** $P(1) \vee P(2) \vee P(3) \vee P(4) \vee P(5)$ **b)** $P(1) \wedge P(2) \wedge P(3) \wedge P(4) \wedge P(5)$ **c)** $\neg(P(1) \vee P(2) \vee P(3) \vee P(4) \vee P(5))$ **d)** $\neg(P(1) \wedge P(2) \wedge P(3) \wedge P(4) \wedge P(5))$ **e)** $(P(1) \wedge P(2) \wedge P(4) \wedge P(5)) \vee (\neg P(1) \vee \neg P(2) \vee \neg P(3) \vee \neg P(4) \vee \neg P(5))$ **21.** São possíveis muitas respostas. **a)** Todos os estudantes no curso de matemática discreta; todos os estudantes no mundo. **b)** Todos os senadores dos Estados Unidos; todos os jogadores de futebol americano universitários. **c)** George W. Bush e Jeb Bush; todos os políticos nos Estados Unidos. **d)** Bill Clinton e George W. Bush; todos os políticos nos Estados Unidos. **23.** Seja $C(x)$ a função proposicional "x está em sua classe". **a)** $\exists x\, H(x)$ e $\exists x(C(x) \wedge H(x))$, em que $H(x)$ é "x sabe falar hindu". **b)** $\forall x\, F(x)$ e $\forall x(C(x) \rightarrow F(x))$, em que $F(x)$ é "x é amigável". **c)** $\exists x \neg B(x)$ e $\exists x(C(x) \wedge \neg B(x))$, em que $B(x)$ é "x nasceu na Califórnia". **d)** $\exists x\, M(x)$ e $\exists x(C(x) \wedge M(x))$, em que $M(x)$ é "x participou de um filme". **e)** $\forall x \neg L(x)$ e $\forall x(C(x) \rightarrow \neg L(x))$, em que $L(x)$ é "x teve um curso de programação lógica". **25.** Seja $P(x)$ "x é perfeito"; seja $F(x)$ "x é seu amigo"; e considere como domínio todas as pessoas. **a)** $\forall x \neg P(x)$ **b)** $\neg \forall x\, P(x)$ **c)** $\forall x(F(x) \rightarrow P(x))$ **d)** $\exists x(F(x) \wedge P(x))$ **e)** $\forall x(F(x) \wedge P(x))$ ou $(\forall x\, F(x)) \wedge (\forall x\, P(x))$ **f)** $(\neg \forall x\, F(x)) \vee (\exists x \neg P(x))$ **27.** Seja $Y(x)$ a função proposicional que x está em sua escola ou classe, conforme apropriado. **a)** Se fizermos $V(x)$ ser "x morou no Vietnã", então temos $\exists x\, V(x)$ se o domínio for apenas seus colegas de escola, ou $\exists x(Y(x) \wedge V(x))$ se o domínio for todas as pessoas. Se $D(x, y)$ significar que a pessoa x morou no país y, então podemos reescrever a última como $\exists x(Y(x) \wedge D(x,$ Vietnã$))$. **b)** Se $H(x)$ for "x fala hindu", então temos $\exists x \neg H(x)$ se o domínio for apenas seus colegas de escola, ou $\exists x(Y(x) \wedge \neg H(x))$ se o domínio for todas as pessoas. Se $S(x, y)$ significar que a pessoa x fala a língua y, então podemos reescrever esta última como $\exists x(Y(x) \wedge \neg S(x,$ hindu$))$. **c)** Se $J(x)$, $P(x)$ e $C(x)$ forem as funções proposicionais que afirmam que x conhece Java, Prolog e C++, respectivamente, então temos $\exists x(J(x) \wedge P(x) \wedge C(x))$ se o domínio for apenas seus colegas de escola, ou $\exists x(Y(x) \wedge J(x) \wedge P(x) \wedge C(x))$ se o domínio for todas as pessoas. Se $K(x, y)$ significar que a pessoa x conhece a linguagem de programação y, então podemos reescrever esta última como $\exists x(Y(x) \wedge K(x,$ Java$) \wedge K(x,$ Prolog$) \wedge K(x,$ C++$))$. **d)** Se $T(x)$ for "x gosta de comida tailandesa", então temos $\forall x\, T(x)$ se o domínio for apenas seus colegas de classe, ou $\forall x(Y(x) \rightarrow T(x))$ se o domínio for todas as pessoas. Se $E(x, y)$ significa que a pessoa x gosta da comida de tipo y, então podemos reescrever esta última como $\forall x(Y(x) \rightarrow E(x,$ tailandesa$))$. **e)** Se $H(x)$ for "x joga hóquei", então temos $\exists x \neg H(x)$ se o domínio for apenas seus colegas de classe, ou $\exists x(Y(x) \wedge \neg H(x))$ se o domínio for todas as pessoas. Se $P(x, y)$ significa que a pessoa x joga o jogo y, então podemos reescrever esta última como $\exists x(Y(x) \wedge \neg P(x,$ hóquei$))$. **29.** Indique por $T(x)$ o fato de x ser uma tautologia e por $C(x)$ o fato de x ser uma contradição. **a)** $\exists x\, T(x)$ **b)** $\forall x(C(x) \rightarrow T(\neg x))$ **c)** $\exists x \exists y(\neg T(x) \wedge \neg C(x) \wedge \neg T(y) \wedge \neg C(y) \wedge T(x \vee y))$ **d)** $\forall x \forall y((T(x) \wedge T(y)) \rightarrow T(x \wedge y))$ **31. a)** $Q(0,0,0) \wedge Q(0,1,0)$ **b)** $Q(0,1,1) \vee Q(1, 1, 1) \vee Q(2, 1, 1)$ **c)** $\neg Q(0, 0, 0) \vee \neg Q(0, 0, 1)$ **d)** $\neg Q(0, 0, 1) \vee \neg Q(1, 0, 1) \vee \neg Q(2,0,1)$ **33. a)** Seja $T(x)$ o predicado que x aprende truques novos, e considere o domínio como sendo cães velhos. O original é $\exists x\, T(x)$. A negação é $\forall x \neg T(x)$: "Nenhum cão velho aprende novos truques." **b)** Seja $C(x)$ o predicado que x sabe cálculo, e considere o domínio como sendo coelhos. O original é $\neg \exists x\, C(x)$. A negação é $\exists x\, C(x)$: "Existe um coelho que sabe cálculo." **c)** Seja $F(x)$ o predicado que x pode voar, e considere o domínio como sendo pássaros. O original é $\forall x\, F(x)$. A negação é $\exists x \neg F(x)$: "Existe um pássaro que não pode voar." **d)** Seja $T(x)$ o predicado que x fala, e considere o domínio como sendo cães. O original é $\neg \exists x\, T(x)$. A negação é $\exists x\, T(x)$: "Existe um cão que fala." **e)** Seja $F(x)$ e $R(x)$ os predicados que x fale francês e russo, respectivamente, e considere o domínio como sendo pessoas em sua sala. O original é $\neg \exists x(F(x) \wedge R(x))$. A negação é $\exists x(F(x) \wedge R(x))$: "Existe alguém em sua sala que fala francês e russo". **35. a)** Não existe contra-exemplo. **b)** $x = 0$ **c)** $x = 2$ **37. a)** $\forall x((F(x, 25.000) \vee S(x, 25)) \rightarrow E(x))$, em que $E(x)$ é "A pessoa x se qualificou como um viajante de elite em um dado ano", $F(x, y)$ é "A pessoa x voa mais de y milhas em um dado ano" e $S(x, y)$ é "A pessoa x pega mais de y vôos em um dado ano" **b)** $\forall x(((M(x) \wedge T(x, 3)) \vee (\neg M(x) \wedge T(x, 3,5))) \rightarrow Q(x))$, em que $Q(x)$ é "A pessoa x se classifica para a maratona", $M(x)$ é "A pessoa x é um homem" e $T(x, y)$ é "A pessoa x correu a maratona em menos de y horas aula" **c)** $M \rightarrow ((H(60) \vee (H(45) \wedge T)) \wedge \forall y\, G(\mathrm{B}, y))$, em que M é a proposição "O estudante recebeu o título de mestre", $H(x)$ é "O estudante freqüentou no mínimo x horas/aula", T é a proposição "O estudante escreveu uma tese" e $G(x, y)$ é "A pessoa teve conceito x ou maior no curso y" **d)** $\exists x((T(x, 21) \wedge G(x, 4,0))$, em que $T(x, y)$ é "A pes-

soa x cursou mais de y créditos" e $G(x, p)$ é "A pessoa x tirou nota média p" (supomos que estamos falando de um dado semestre) **39. a)** Se existir uma impressora que esteja tanto quebrada quanto ocupada, então algum trabalho foi perdido. **b)** Se toda impressora estiver ocupada, então existe um trabalho na fila. **c)** Se existir um trabalho que esteja tanto na fila quanto perdido, então alguma impressora está quebrada. **d)** Se toda impressora estiver ocupada e todo trabalho estiver na fila, então algum trabalho foi perdido. **41. a)** $(\exists x\, F(x, 10)) \rightarrow \exists x\, S(x)$, em que $F(x, y)$ é "O disco x tem mais de y kilobytes de espaço livre" e $S(x)$ é "O e-mail x pode ser salvo" **b)** $(\exists x\, A(x)) \rightarrow \forall x(Q(x) \rightarrow T(x))$, em que $A(x)$ é "O alerta x está ativado", $Q(x)$ é "A mensagem x está na fila" e $T(x)$ é "A mensagem x foi transmitida" **c)** $\forall x((x \neq \text{console principal}) \rightarrow T(x))$, em que $T(x)$ é "A tela de diagnóstico rastreia o *status* do sistema x" **d)** $\forall x(\neg L(x) \rightarrow B(x))$, em que $L(x)$ é "O anfitrião da videoconferência pôs x em uma lista especial" e $B(x)$ é "O participante x recebeu uma conta" **43.** Elas não são equivalentes. Seja $P(x)$ qualquer função proposicional que seja às vezes verdadeira e às vezes falsa, e seja $Q(x)$ qualquer função proposicional que seja sempre falsa. Então $\forall x(P(x) \rightarrow Q(x))$ é falsa, mas $\forall x\, P(x) \rightarrow \forall x\, Q(x)$ é verdadeira. **45.** Ambas as afirmações são verdadeiras precisamente quando pelo menos uma entre $P(x)$ e $Q(x)$ for verdadeira para pelo menos um valor de x no domínio. **47. a)** Se A for verdadeira, então ambos os lados são logicamente equivalentes a $\forall x\, P(x)$. Se A for falsa, o lado esquerdo é claramente falso. Além disso, para todo x, $P(x) \wedge A$ é falsa, de modo que o lado direito é falso. Logo, os dois lados são logicamente equivalentes. **b)** Se A for verdadeira, então ambos os lados são logicamente equivalentes a $\exists x\, P(x)$. Se A for falsa, o lado esquerdo é claramente falso. Além disso, para todo x, $P(x) \wedge A$ é falsa, de modo que $\exists x(P(x) \wedge A)$ é falsa. Logo, os dois lados são logicamente equivalentes. **49.** Podemos estabelecer estas equivalências argumentando que um lado é verdadeiro se e somente se o outro lado for verdadeiro. **a)** Suponha que A seja verdadeira. Então, para cada x, $P(x) \rightarrow A$ é verdadeira; portanto, o lado esquerdo é sempre verdadeiro neste caso. Por um raciocínio análogo, o lado direito é sempre verdadeiro neste caso. Portanto, as duas proposições são logicamente equivalentes no caso em que A é verdadeira. Por outro lado, suponha que A seja falsa. Existem dois subcasos. Se $P(x)$ for falsa para todo x, então $P(x) \rightarrow A$ é trivialmente verdadeira, de modo que o lado esquerdo é trivialmente verdadeiro. O mesmo raciocínio mostra que o lado direito também é verdadeiro, pois neste subcaso $\exists x\, P(x)$ é falso. Para o segundo subcaso, suponha que $P(x)$ seja verdadeira para algum x. Então, para cada x, $P(x) \rightarrow A$ é falsa, de modo que o lado esquerdo é falso. O lado direito também é falso, pois neste subcaso $\exists x\, P(x)$ é verdadeira, mas A é falsa. Assim, em todos os casos, as duas proposições têm o mesmo valor-verdade. **b)** Se A for verdadeira, então ambos os lados são trivialmente verdadeiros, pois as afirmações condicionais têm conclusões verdadeiras. Se A for falsa, então existem dois subcasos. Se $P(x)$ for falsa para algum x, então $P(x) \rightarrow A$ é trivialmente verdadeira para aquele x, de modo que o lado esquerdo é verdadeiro. O mesmo raciocínio mostra que o lado direito também é verdadeiro, pois neste subcaso $\forall x\, P(x)$ é falsa. Para o segundo subcaso, suponha que $P(x)$ seja verdadeira para todo x. Então, para todo x, $P(x) \rightarrow A$ é falsa, de modo que o lado esquerdo é falso (não existe nenhum x que torne a afirmação condicional verdadeira). O lado direito também é falso, pois é uma afirmação condicional com uma hipótese verdadeira e uma conclusão falsa. Logo, em todos os casos, as duas proposições têm o mesmo valor-verdade. **51.** Para mostrar que estas não são logicamente equivalentes, seja $P(x)$ a afirmação "x é positivo" e seja $Q(x)$ a afirmação "x é negativo" com domínio no conjunto dos inteiros. Então $\exists x\, P(x) \wedge \exists x\, Q(x)$ é verdadeira, mas $\exists x(P(x) \wedge Q(x))$ é falsa. **53. a)** Verdadeiro **b)** Falso, a menos que o domínio consista em apenas um elemento. **c)** Verdadeiro **55. a)** Sim **b)** Não **c)** juana, kiko **d)** math273, cs301 **e)** juana, kiko **57.** `irmaos(X,Y):-mae(M,X), mae(M,Y), pai(F,X), pai(F,Y)` **59. a)** $\forall x(P(x) \rightarrow \neg Q(x))$ **b)** $\forall x(Q(x) \rightarrow R(x))$ **c)** $\forall x(P(x) \rightarrow \neg R(x))$ **d)** A conclusão não é válida. Podem existir professores vãos, mas as hipóteses não excluem a possibilidade de que existem outras pessoas vãs além das ignorantes. **61. a)** $\forall x(P(x) \rightarrow \neg Q(x))$ **b)** $\forall x(R(x) \rightarrow \neg S(x))$ **c)** $\forall x(\neg Q(x) \rightarrow S(x))$ **d)** $\forall x(P(x) \rightarrow \neg R(x))$ **e)** A conclusão é válida. Suponha que x seja um bebê. Então, pela primeira hipótese, x é ilógico, e pela terceira hipótese, x é desprezível. A segunda hipótese diz que se x pudesse controlar um crocodilo, então x não seria desprezível. Portanto, x não pode controlar um crocodilo.

Seção 1.4

1. a) Para todo número real x existe um número real y tal que x é menor que y. **b)** Para todo número real x e todo número real y, se x e y forem ambos não negativos, então seu produto é não negativo. **c)** Para todo número real x e todo número real y, existe um número real z tal que $xy = z$. **3. a)** Existe um estudante de sua sala que enviou um e-mail para um estudante de sua sala. **b)** Existe um estudante de sua sala que enviou um e-mail para todo estudante de sua sala. **c)** Todo estudante de sua sala mandou um e-mail para, pelo menos, um estudante de sua sala. **d)** Existe um estudante em sua sala para quem foi enviado um e-mail por todo estudante de sua sala. **e)** Todo estudante de sua sala recebeu pelo menos um e-mail de pelo menos um estudante de sua sala. **f)** Todo estudante em sua sala enviou um e-mail para todo estudante da sala. **5. a)** Sarah Smith visitou www.att.com. **b)** Pelo menos uma pessoa visitou www.imdb.org. **c)** Jose Orez visitou pelo menos um site da Web. **d)** Existe um site da Web que ambos, Ashok Puri e Cindy Yoon, visitaram. **e)** Existe uma pessoa além de David Belcher que visitou todos os sites da Web que David Belcher visitou. **f)** Existem duas pessoas diferentes que visitaram exatamente os mesmos sites da Web. **7. a)** Abdallah Hussein não gosta da cozinha japonesa. **b)** Alguns estudantes em sua escola gostam de cozinha coreana e todos em sua escola gostam da cozinha mexicana. **c)** Existe alguma cozinha de que ou Monique Arsenault ou Jay Johnson gosta. **d)** Para todo par de estudantes distintos em sua escola, existe alguma cozinha de que pelo menos um deles não goste. **e)** Existem dois estudantes em sua escola que gostam exatamente das mesmas cozinhas. **f)** Para todo par de estudantes distintos em sua escola, existe alguma cozinha sobre a qual eles têm a mesma opinião (ou ambos gostam dela ou ambos não gostam dela). **9. a)** $\forall x\, L(x, \text{Jerry})$ **b)** $\forall x \exists y\, L(x, y)$ **c)** $\exists y \forall x\, L(x, y)$ **d)** $\forall x \exists y \neg L(x, y)$ **e)** $\exists x \neg L(\text{Lydia}, x)$ **f)** $\exists x\, \forall y \neg L(y, x)$ **g)** $\exists x(\forall y\, L(y, x) \wedge \forall z((\forall w\, L(w, z)) \rightarrow z =$

x)) **h)** $\exists x \exists y (x \neq y \wedge L(\text{Lynn}, x) \wedge L(\text{Lynn}, y) \wedge \forall z (L(\text{Lynn}, z) \rightarrow (z = x \vee z = y)))$ **i)** $\forall x\, L(x, x)$ **j)** $\exists x\, \forall y (L(x, y) \leftrightarrow x = y)$ **11. a)** A(Lois, professor Michaels) **b)** $\forall x(S(x) \rightarrow A(x, \text{professor Gross}))$ **c)** $\forall x(F(x) \rightarrow (A(x, \text{professor Miller}) \vee A(\text{professor Miller}, x)))$ **d)** $\exists x(S(x) \wedge \forall y(F(y) \rightarrow \neg A(x, y)))$ **e)** $\exists x(F(x) \wedge \forall y(S(y) \rightarrow \neg A(y, x)))$ **f)** $\forall y(F(y) \rightarrow \exists x(S(x) \vee A(x, y)))$ **g)** $\exists x(F(x) \wedge \forall y((F(y) \wedge (y \neq x)) A(x, y)))$ **h)** $\exists x(S(x) \wedge \forall y(F(y) \rightarrow \neg A(y, x)))$ **13. a)** $\neg M$(Chou, Koko) **b)** $\neg M(\text{Arlene}, \text{Sarah}) \wedge \neg T(\text{Arlene}, \text{Sarah})$ **c)** $\neg M$(Deborah, Jose) **d)** $\forall x M(x,$ Ken) **e)** $\forall x \neg T$ (x, Nina) **f)** $\forall x(T\text{-}x, \text{Avi}) \vee M(x, \text{Avi}))$ **g)** $\exists x \forall y (y \neq x \rightarrow M(x, y))$ **h)** $\exists x \forall y (y \neq x \rightarrow (M(x, y) \vee T(x, y)))$ **i)** $\exists x \exists y (x \neq y \wedge M(x, y) \wedge M(y, x))$ **j)** $\exists x M(x, x)$ **k)** $\exists x \forall y\, (x \neq y \rightarrow (\neg M(x, y) \wedge \neg T(y, x)))$ **l)** $\forall x (\exists y (x \neq y \wedge (M(y, x) \vee T(y, x))))$ **m)** $\exists x \exists y (x \neq y \wedge M(x, y) \wedge T(y, x))$ **n)** $\exists x \exists y (x \neq y \wedge \forall z((z \neq x \wedge z \neq y) \rightarrow (M(x, z) \vee M(y, z) \vee T(x, z) \vee T(y, z))))$ **15. a)** $\forall x\, P(x)$, em que $P(x)$ é "x precisa de um curso de matemática discreta" e o domínio consiste em todos os estudantes de ciência da computação. **b)** $\exists x\, P(x)$, em que $P(x)$ é "x possui seu próprio computador" e o domínio consiste em todos os estudantes nesta sala. **c)** $\forall x \exists y P(x, y)$, em que $P(x, y)$ é "x participou do curso y", o domínio para x consiste em todos os estudantes nesta sala, e o domínio para y consiste em todos os cursos de ciência da computação. **d)** $\exists x \exists y P(x, y)$, em que $P(x, y)$ e domínios são os mesmos da parte (c). **e)** $\forall x \forall y P(x, y)$, em que $P(x, y)$ é "x já esteve em y", o domínio para x consiste em todos os estudantes desta sala, e o domínio para y consiste em todos os prédios do campus. **f)** $\exists x \exists y \forall z (P(z, y) \rightarrow Q(x, z))$, em que $P(z, y)$ é "z está em y" e $Q(x, z)$ é "x já esteve em z"; o domínio para x consiste em todos os estudantes na sala, o domínio para y consiste em todos os prédios do campus, e o domínio para z consiste em todas as salas. **g)** $\forall x \forall y \exists z (P(z, y) \wedge Q(x, z))$, com o mesmo ambiente que na parte (f) **17. a)** $\forall u \exists m (A(u, m) \wedge \forall n (n \neq m \rightarrow \neg A(u, n)))$, em que $A(u, m)$ significa que o usuário u tem acesso à caixa de entrada m **b)** $\exists p \forall e (H(e) \wedge S(p,$ executando$)) \rightarrow S$ (kernel, funcionando corretamente), em que $H(e)$ significa que a condição de erro e está em vigor e $S(x, y)$ significa que o *status* de x é y **c)** $\forall u \forall s (E(s,$ .edu$) \rightarrow A(u, s))$, em que $E(s, x)$ significa que o site da Web s tem a extensão x, e $A(u, s)$ significa que o usuário u pode acessar o site da Web s **d)** $\exists x \exists y (x \neq y \wedge \forall z ((\forall s\, M(z, s)) \leftrightarrow (z = x \vee z = y)))$, em que $M(a, b)$ significa que o sistema a monitora o servidor remoto b **19. a)** $\forall x \forall y ((x < 0) \wedge (y < 0) \rightarrow (x + y < 0))$ **b)** $\neg \forall x \forall y ((x > 0) \wedge (y > 0) \rightarrow (x - y > 0))$ **c)** $\forall x \forall y\, (x^2 + y^2 \geq (x + y)^2)$ **d)** $\forall x \forall y\ (|xy| = |x||y|)$ **21.** $\forall x \exists a \exists b \exists c \exists d\, ((x > 0) \rightarrow x = a^2 + b^2 + c^2 + d^2)$, em que o domínio consiste em todos os inteiros **23. a)** $\forall x \forall y\, ((x < 0) \wedge (y < 0) \rightarrow (xy > 0))$ **b)** $\forall x (x - x = 0)$ **c)** $\forall x \exists a \exists b (a \neq b \wedge \forall c (c^2 = x \leftrightarrow (c = a \vee c = b)))$ **d)** $\forall x((x < 0) \rightarrow \neg \exists y (x = y^2))$ **25. a)** Existe um elemento neutro da multiplicação para os reais. **b)** O produto de dois números reais negativos é sempre um número real positivo. **c)** Existem números reais x e y tais que x^2 excede y, mas x é menor do que y. **d)** Os números reais são fechados pela operação de adição. **27. a)** Verdadeira **b)** Verdadeira **c)** Verdadeira **d)** Verdadeira **e)** Verdadeira **f)** Falsa **g)** Falsa **h)** Verdadeira **i)** Falsa **29. a)** $P(1, 1) \wedge P(1, 2) \wedge P(1, 3) \wedge P(2, 1) \wedge P(2, 2) \wedge P(2, 3) \wedge P(3, 1) \wedge P(3, 2) \wedge P(3, 3)$ **b)** $P(1, 1) \vee P(1, 2) \vee P(1, 3) \vee P(2, 1) \vee P(2, 2) \vee P(2, 3) \vee P(3, 1) \vee P(3, 2) \vee P(3, 3)$ **c)** $(P(1, 1) \wedge P(1, 2) \wedge P(1, 3)) \vee (P(2, 1) \wedge P(2, 2) \wedge P(2, 3)) \vee (P(3, 1) \wedge P(3, 2) \wedge P(3, 3))$ **d)** $(P(1, 1) \vee P(2, 1) \vee P(3, 1)) \wedge (P(1, 2) \vee P(2, 2) \vee P(3, 2)) \wedge (P(1, 3) \vee P(2, 3) \vee P(3, 3))$ **31. a)** $\exists x \forall y \exists z \neg T(x, y, z)$ **b)** $\exists x \forall y \neg P(x, y) \wedge \exists x \forall y \neg Q(x, y)$ **c)** $\exists x \forall y\, (\neg P(x, y) \vee \forall z \neg R(x, y, z))$ **d)** $\exists x \forall y (P(x, y) \wedge \neg Q(x, y))$ **33. a)** $\exists x \exists y \neg P(x, y)$ **b)** $\exists y \forall x \neg P(x, y)$ **c)** $\exists y \exists x (\neg P(x, y) \wedge \neg Q(x, y))$ **d)** $(\forall x \forall y P(x, y)) \vee (\exists x \exists y \neg Q(x, y))$ **e)** $\exists x (\forall y \exists z \neg P(x, y, z) \vee \forall z \exists y \neg P(x, y, z))$ **35.** Qualquer domínio com quatro ou mais membros torna a afirmação verdadeira; qualquer domínio com três ou menos membros torna a afirmação falsa. **37. a)** Existe alguém nesta sala tal que para quaisquer dois cursos de matemática diferentes, estes não são os dois únicos cursos de matemática que a pessoa freqüentou. **b)** Qualquer pessoa ou visitou a Líbia ou não visitou um pais diferente da Líbia. **c)** Alguém escalou todas as montanhas do Himalaia. **d)** Existe alguém que nunca participou de um filme ao lado de Kevin Bacon nem contracenou com alguém que já filmou com Kevin Bacon. **39. a)** $x = 2$, $y = -2$ **b)** $x = -4$ **c)** $x = 17, y = -1$ **41.** $\forall x \forall y \forall z ((x \cdot y) \cdot z = x \cdot (y \cdot z))$ **43.** $\forall m \forall b (m \neq 0 \rightarrow \exists x (mx + b = 0 \wedge \forall w (mw + b = 0 \rightarrow w = x)))$ **45. a)** Verdadeira **b)** Falsa **c)** Verdadeira **47.** $\neg(\exists x \forall y P(x, y)) \leftrightarrow \forall x(\neg \forall y P(x, y)) \leftrightarrow \forall x \exists y \neg P(x, y)$ **49. a)** Suponha que $\forall x\, P(x) \wedge \exists x Q(x)$ seja verdadeira. Então, $P(x)$ é verdadeira para todo x e existe um elemento y para o qual $Q(y)$ é verdadeira. Como $P(x) \wedge Q(y)$ é verdadeira para todo x e existe um y para o qual $Q(y)$ é verdadeira, $\forall x \exists y (P(x) \wedge Q(y))$ é verdadeira. Reciprocamente, suponha que a segunda proposição seja verdadeira. Seja x um elemento no domínio. Existe um y tal que $Q(y)$ é verdadeira, de modo que $\exists x Q(x)$ é verdadeira. Como $\forall x\, P(x)$ também é verdadeira, segue que a primeira proposição é verdadeira. **b)** Suponha que $\forall x\, P(x) \vee \exists x Q(x)$ seja verdadeira. Então, ou $P(x)$ é verdadeira para todo x, ou existe um y para o qual $Q(y)$ é verdadeira. No primeiro caso, $P(x) \vee Q(y)$ é verdadeira para todo x, de modo que $\forall x \exists y (P(x) \vee Q(y))$ é verdadeira. No segundo caso, $Q(y)$ é verdadeira para um y particular, de modo que $P(x) \vee Q(y)$ é verdadeira para todo x e, conseqüentemente, $\forall x \exists y (P(x) \vee Q(y))$ é verdadeira. Reciprocamente, suponha que a segunda proposição seja verdadeira. Se $P(x)$ for verdadeira para todo x, então a primeira proposição é verdadeira. Caso contrário, $P(x)$ é falsa para algum x, e para este x deve existir um y tal que $P(x) \vee Q(y)$ é verdadeira. Portanto, $Q(y)$ deve ser verdadeira, de modo que $\exists y Q(y)$ é verdadeira. Segue que a primeira proposição deve ser válida. **51.** Mostraremos como uma expressão pode ser posta na forma normal prenex (FNP) se subexpressões nela puderem ser colocadas na FNP. Então, trabalhando de dentro para fora, qualquer expressão pode ser posta em FNP. (Para formalizar o argumento, é necessário usar o método de indução estrutural que será discutido na Seção 4.3.) Pelo Exercício 45 da Seção 1.2, podemos supor que a proposição usa apenas $\vee$ e $\neg$ como conectores lógicos. Observe agora que qualquer expressão sem quantificadores já está em FNP. (Este é o caso base da argumentação.) Suponha agora que a proposição seja da forma $Qx\, P(x)$, em que Q é um quantificador. Como $P(x)$ é uma expressão mais curta que a proposição original, podemos colocá-la em FNP. Então, Qx seguida por esta PNF está novamente em FNP e é equivalente à proposição original. A seguir, suponha que a proposição seja da forma $\neg P$. Se P já estiver em FNP, passamos o sinal de negação por todos os quantificadores usando as equivalências da Tabela 2 da Seção 1.3. Finalmente, suponha que a proposição seja da forma $P \vee$

Q, em que cada um entre P e Q está na FNP. Se apenas uma entre P e Q tiver quantificadores, então podemos usar o Exercício 46 da Seção 1.3 para trazer o quantificador para frente de ambas. Se ambas, P e Q, tiverem quantificadores, podemos usar o Exercício 45 da Seção 1.3, o Exercício 48, ou a parte (b) do Exercício 49 para reescrever $P \vee Q$ com dois quantificadores precedendo a disjunção de uma proposição da forma $R \vee S$, e então pôr $R \vee S$ em FNP.

Seção 1.5

1. *Modus ponens*; válido; a conclusão é verdadeira, pois as hipóteses são verdadeiras. **3. a)** Adição **b)** Simplificação **c)** *Modus ponens* **d)** *Modus tollens* **e)** Silogismo hipotético **5.** Seja w "Randy trabalha muito," seja d "Randy é um garoto estúpido," e seja j "Randy conseguirá o emprego". As hipóteses são w, $w \rightarrow d$ e $d \rightarrow \neg j$. Usando *modus ponens* e as primeiras duas hipóteses, segue d. Usando *modus ponens* e a última hipótese, segue $\neg j$, "Randy não conseguirá o emprego", que é a conclusão desejada. **7.** Instanciação universal é usada para concluir que "Se Sócrates for um homem, então Sócrates é mortal". *Modus ponens* é então usado para concluir que Sócrates é mortal. **9. a)** As conclusões válidas são "Eu não tirei folga na terça-feira", "Eu tirei folga na quinta-feira", "Choveu na quinta-feira". **b)** "Eu não comi comida apimentada e não trovejou" é uma conclusão válida. **c)** "Eu sou esperto" é uma conclusão válida. **d)** "Ralph não estuda ciência da computação" é uma conclusão válida. **e)** "Que você compre muitas coisas é bom para os Estados Unidos e é bom para você" é uma conclusão válida. **f)** "Os ratos roem sua própria comida" e "Coelhos não são roedores" são conclusões válidas. **11.** Suponha que $p_1, p_2, ..., p_n$ sejam verdadeiras. Queremos estabelecer que $q \rightarrow r$ é verdadeira. Se q for falsa, então acabamos, trivialmente. Caso contrário, se q for verdadeira, então, pela validade da forma do argumento dado (que sempre que $p_1, p_2, ..., p_n, q$ forem verdadeiras, então r deve ser verdadeira), sabemos que r é verdadeira. **13. a)** Sejam $c(x)$ "x está nesta sala", $j(x)$, "x sabe como escrever programas em JAVA" e $h(x)$, "x pode conseguir um emprego bem remunerado", As premissas são c(Doug), j(Doug), $\forall x(j(x) \rightarrow h(x))$. Usando instanciação universal e a última premissa, segue que j(Doug) $\rightarrow h$(Doug). Aplicando *modus ponens* a esta conclusão e à segunda premissa, segue h(Doug). Usando conjunção e a primeira premissa, segue c(Doug) $\wedge h$(Doug). Finalmente, usando generalização existencial, segue a conclusão desejada, $\exists x(c(x) \wedge h(x))$. **b)** Sejam $c(x)$ "x está na sala," $w(x)$, "x gosta de ver baleias" e $p(x)$, "x se preocupa com a poluição no mar". As premissas são $\exists x(c(x) \wedge w(x))$ e $\forall x(w(x) \rightarrow p(x))$. A partir da primeira premissa, $c(y) \wedge w(y)$ para uma pessoa particular y. Usando simplificação, segue $w(y)$. Usando a segunda premissa e instanciação universal, segue $w(y) \rightarrow p(y)$. Usando *modus ponens*, segue $p(y)$ e, por conjunção, segue $c(y) \wedge p(y)$. Finalmente, por generalização existencial, segue a conclusão desejada, $\exists x(c(x) \wedge p(x))$. **c)** Sejam $c(x)$ "x está na sala", $p(x)$, "x possui um PC" e $w(x)$, "x pode usar um programa de processamento de palavras". AS premissas são c(Zeke),$\forall x(c(x) \rightarrow p(x))$ e $\forall x(\, p(x) \rightarrow w(x))$. Usando a segunda premissa e instanciação universal, segue c(Zeke) $\rightarrow p$(Zeke). Usando a primeira premissa e *modus ponens*, segue p(Zeke). Usando a terceira premissa e instanciação universal, segue p(Zeke) $\rightarrow$ w(Zeke). Finalmente, usando *modus ponens*, segue w(Zeke), a conclusão desejada. **d)** Sejam $j(x)$ "x está em Nova Jersey", $f(x)$ "x vive a 50 milhas do oceano" e $s(x)$, "x já viu o oceano". As premissas são $\forall x(j(x) \rightarrow f(x))$ e $\exists x(j(x) \wedge \neg s(x))$. A segunda hipótese e instanciação existencial implicam que $j(y) \wedge \neg s(y)$ para uma pessoa particular y. Por simplificação, $j(y)$ para esta pessoa y. Usando instanciação universal e a primeira premissa, $j(y) \rightarrow f(y)$, e por *modus ponens*, segue $f(y)$. Por simplificação, $\neg s(y)$ segue de $j(y) \wedge \neg s(y)$. Assim, $f(y) \wedge \neg s(y)$ segue por conjunção. Finalmente, a conclusão desejada, $\exists x(\, f(x) \wedge \neg s(x))$, segue por generalização existencial. **15. a)** Correto, usando instanciação universal e *modus ponens* **b)** Inválido; falácia de afirmar a conclusão **c)** Inválido; falácia de negar a hipótese **d)** Correto, usando instanciação universal e *modus tollens* **17.** Sabemos que existe *algum* x que torna $H(x)$ verdadeira, mas não podemos concluir que Lola é tal x. **19. a)** Falácia de afirmação da conclusão **b)** Falácia de carregar a pergunta **c)** Argumento válido usando *modus tollens* **d)** Falácia de negar a hipótese **21.** Pela segunda premissa, existem alguns leões que não bebem café. Seja Leo uma dessas criaturas. Por simplificação, sabemos que Leo é um leão. Por *modus ponens*, sabemos a partir da primeira premissa que Leo é feroz. Portanto, Leo é feroz e não bebe café. Pela definição de quantificador existencial, existem criaturas ferozes que não bebem café, ou seja, algumas criaturas ferozes não bebem café. **23.** O erro ocorre no passo (5), pois não podemos supor, como foi feito aqui, que o c que torna P verdadeira é o mesmo c que torna Q verdadeira. **25.** Nos foram dadas as premissas $\forall x(P(x) \rightarrow Q(x))$ e $\neg Q(a)$. Queremos mostrar que $\neg P(a)$. Suponha, pelo contrário, que $\neg P(a)$ não seja verdadeira. Então, $P(a)$ é verdadeira. Portanto, pelo *modus ponens* universal, temos $Q(a)$. Mas isso contradiz a premissa dada $\neg Q(a)$. Portanto, nossa hipótese deve estar errada, e assim $\neg P(a)$ é verdadeira, como desejado.

27.

Passo	**Razão**
1. $\forall x(P(x) \wedge R(x))$	Premissa
2. $P(a) \wedge R(a)$	Instanciação universal a partir de (1)
3. $P(a)$	Simplificação a partir de (2)
4. $\forall x(P(x) \rightarrow (Q(x) \wedge S(x)))$	Premissa
5. $Q(a) \wedge S(a)$	*Modus ponens* universal a partir de (3) e (4)
6. $S(a)$	Simplificação a partir de (5)
7. $R(a)$	Simplificação a partir de (2)
8. $R(a) \wedge S(a)$	Conjunção a partir de (7) e (6)
9. $\forall x(R(x) \wedge S(x))$	Generalização universal a partir de (5)

29.

Passo	**Razão**
1. $\exists x \neg P(x)$	Premissa
2. $\neg P(c)$	Instanciação existencial a partir de (1)
3. $\forall x(P(x) \vee Q(x))$	Premissa
4. $P(c) \vee Q(c)$	Instanciação universal a partir de (3)
5. $Q(c)$	Silogismo disjuntivo a partir de (4) e (2)
6. $\forall x(\neg Q(x) \vee S(x))$	Premissa

7. $\neg Q(c) \vee S(c)$	Instanciação universal a partir de (6)
8. $S(c)$	Silogismo disjuntivo a partir de (5) e (7)
9. $\forall x(R(x) \rightarrow \neg S(x))$	Premissa
10. $R(c) \rightarrow \neg S(c)$	Instanciação universal a partir de (9)
11. $\neg R(c)$	*Modus tollens* a partir de (8) e (10)
12. $\exists x \neg R(x)$	Generalização existencial a partir de (11)

31. Seja p "Está chovendo"; seja q "Ivete tem sua sombrinha"; seja r "Ivete se molha." As hipóteses são $\neg p \vee q$, $\neg q \vee \neg r$, e $p \vee \neg r$. A resolução das duas primeiras dá $\neg p \vee \neg r$. A resolução desta e da terceira hipótese dá $\neg r$, como desejado. **33.** Suponha que esta proposição seja satisfatória. Usar a resolução nas primeiras duas cláusulas nos permite concluir $q \vee q$; em outras palavras, sabemos que q tem de ser verdadeira. Usar a resolução nas duas últimas cláusulas nos permite concluir $\neg q \vee \neg q$; em outras palavras, sabemos que $\neg q$ deve ser verdadeira. Isso é uma contradição. Assim, a proposição não é satisfatória. **35.** Válido.

Seção 1.6

1. Sejam $n = 2k + 1$ e $m = 2l + 1$ inteiros ímpares. Então, $n + m = 2(k + l + 1)$ é par. **3.** Suponha que n seja par. Então, $n = 2k$ para algum inteiro k. Portanto, $n^2 = (2k)^2 = 4k^2 = 2(2k^2)$. Como escrevemos n^2 como 2 vezes um inteiro, concluímos que n^2 é par. **5.** Demonstração direta: suponha que $m + n$ e $n + p$ sejam pares. Então, $m + n = 2s$ para algum inteiro s e $n + p = 2t$ para algum inteiro t. Se somarmos estas expressões, obtemos $m + p + 2n = 2s + 2t$. Subtraindo $2n$ de ambos os lados e fatorando, temos $m + p = 2s + 2t - 2n = 2(s + t - n)$. Como escrevemos $m + p$ como 2 vezes um inteiro, concluímos que $m + p$ é par. **7.** Como n é ímpar, podemos escrever $n = 2k + 1$ para algum inteiro k. Então, $(k + 1)^2 - k^2 = k^2 + 2k + 1 - k^2 = 2k + 1 = n$. **9.** Suponha que r seja racional e que i seja irracional e que $s = r + i$ seja racional. Então, pelo Exemplo 7, $s + (-r) = i$ é racional, o que é uma contradição. **11.** Como $\sqrt{2} \cdot \sqrt{2} = 2$ é racional e $\sqrt{2}$ é irracional, o produto de dois números irracionais não é necessariamente irracional. **13.** Demonstração por contraposição: se $1/x$ fosse racional, então, por definição, $1/x = p/q$ para alguns inteiros p e q com $q \neq 0$. Como $1/x$ não pode ser 0 (se fosse, então teríamos a contradição $1 = x \cdot 0$ ao multiplicar ambos os lados por x), sabemos que $p \neq 0$. Agora, $x = 1/(1/x) = 1/(p/q) = q/p$ pelas regras usuais da álgebra e da aritmética. Portanto, x pode ser escrito como o quociente de dois inteiros com o denominador diferente de zero. Logo, por definição, x é racional. **15.** Suponha que não seja verdade que $x \geq 1$ ou $y \geq 1$. Então, $x < 1$ e $y < 1$. Somando estas duas desigualdades, obtemos $x + y < 2$, que é a negação de $x + y \geq 2$. **7. a)** Suponha que n seja ímpar, de modo que $n = 2k + 1$ para algum inteiro k. Então, $n^3 + 5 = 2(4k^3 + 6k^2 + 3k + 3)$. Como $n^3 + 5$ é duas vezes algum inteiro, ele é par. **b)** Suponha que $n^3 + 5$ seja ímpar e que n seja ímpar. Como n é ímpar e o produto de dois números ímpares é ímpar, segue que n^2 é ímpar e, então, que n^3 é ímpar. Mas, então, $5 = (n^3 + 5) - n^3$ teria de ser par, pois é a diferença entre dois números ímpares. Portanto, a suposição que $n^3 + 5$ e n eram ambos ímpares é falsa. **19.** A proposição é vagamente verdadeira, pois 0 não é um inteiro positivo. Demonstração por vacuidade. **21.** $P(1)$ é verdadeira pois $(a + b)^1 = a + b \geq a^1 + b^1 = a + b$. Demonstração direta. **23.** Se escolhêssemos 9 ou menos dias em cada dia da semana, isto cobriria no máximo $9 \cdot 7 = 63$ dias. Mas escolhemos 64 dias. Esta contradição mostra que pelo menos 10 dias dias que escolhemos devem ser no mesmo dia da semana. **25.** Suponha por contradição que a/b seja uma raiz racional, em que a e b são inteiros e esta fração está o mais simplificada possível (isto é, a e b não tem divisores comuns maiores do que 1). Substitua a raiz proposta na equação para obter $a^3/b^3 + a/b + 1 = 0$. Multiplique por b^3 para obter $a^3 + ab^2 + b^3 = 0$. Se a e b forem ambos ímpares, então o lado esquerdo é a soma de três números e, portanto, deve ser ímpar. Se a for ímpar e b for par, então o lado esquerdo é ímpar + par + par, o que é novamente ímpar. Analogamente, se a for par e b for ímpar, então o lado esquerdo é par + par + ímpar, o que é novamente ímpar. Como a fração a/b está simplificada, não pode ocorrer de ambos, a e b, serem pares. Portanto, em todos os casos, o lado esquerdo é ímpar e, portanto, não pode ser igual a 0. Esta contradição mostra que não existe nenhuma raiz. **27.** Primeiro, suponha que n seja ímpar, de modo que $n = 2k + 1$ para algum inteiro k. Então, $5n + 6 = 5(2k + 1) + 6 = 10k + 11 = 2(5k + 5) + 1$. Logo, $5n + 6$ é ímpar. Para demonstrar a recíproca, suponha que n seja par, de modo que $n = 2k$ para algum inteiro k. Então, $5n + 6 = 10k + 6 = 2(5k + 3)$, de modo que $5n + 6$ é par. Assim, n é ímpar se e somente se $5n + 6$ for ímpar. **29.** Esta proposição é verdadeira. Suponha que m não seja nem 1 nem -1. Então, mn tem um fator m maior do que 1. Por outro lado, $mn = 1$, e 1 não tem um fator assim. Logo, $m = 1$ ou $m = -1$. No primeiro caso, $n = 1$, e no segundo caso, $n = -1$, pois $n = 1/m$. **31.** Demonstramos que todas estas são equivalentes a x ser par. Se x for par, então $x = 2k$ para algum inteiro k. Portanto, $3x + 2 = 3 \cdot 2k + 2 = 6k + 2 = 2(3k + 1)$, que é par, pois foi escrito na forma $2t$, em que $t = 3k + 1$. Analogamente, $x + 5 + 2k + 5 = 2k + 4 + 1 = 2(k + 2) + 1$, de modo que $x + 5$ é ímpar; e $x^2 = (2k)^2 = 2(2k^2)$, de modo que x é par. Para as recíprocas, usaremos uma demonstração por contraposição. Assim, suponha que x não seja par; logo, x é ímpar e podemos escrever $x = 2k + 1$ para algum inteiro k. Então, $3x + 2 = 3(2k + 1) + 2 = 6k + 5 = 2(3k + 2) + 1$, que é ímpar (isto é, não é par), pois foi escrito na forma $2t + 1$, em que $t = 3k + 2$. Analogamente, $x + 5 = 2k + 1 + 5 = 2(k + 3)$, de modo que $x + 5$ é par (isto é, não é ímpar). Que x^2 é ímpar já foi demonstrado no Exemplo 1. **33.** Damos demonstrações por contraposição de $(i) \rightarrow (ii)$, $(ii) \rightarrow (i)$, $(i) \rightarrow (iii)$ e $(iii) \rightarrow (i)$. Para a primeira delas, suponha que $3x + 2$ seja racional, ou seja, igual a p/q para inteiros p e q com $q \neq 0$. Então, podemos escrever $x = ((p/q) - 2)/3 = (p - 2q)/(3q)$, em que $3q \neq 0$. Isso mostra que x é racional. Para a segunda afirmação condicional, suponha que x seja racional, ou seja, igual a p/q com $q \neq 0$. Assim, podemos escrever $3x + 2 = (3p + 2q)/q$, em que $q \neq 0$. Isto mostra que $3x + 2$ é racional. Para a terceira afirmação condicional, suponha que $x/2$ seja racional, ou seja, igual a p/q para inteiros p e q com $q \neq 0$. Então, podemos escrever $x = 2$

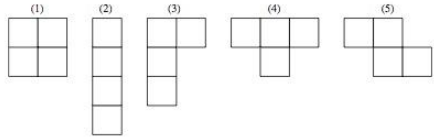
p/q, em que $q \neq 0$. Isso mostra que x é racional. E para a quarta afirmação condicional, suponha que x seja racional, ou seja, igual a p/q para inteiros p e q com $q \neq 0$. Então, podemos escrever $x/2 = p/(2q)$, em que $2q \neq 0$. Isso mostra que $x/2$ é racional. **35.** Não **37.** Suponha que $p_1 \rightarrow p_4 \rightarrow p_2 \rightarrow p_5 \rightarrow p_3 \rightarrow p_1$. Para demonstrar que uma destas proposições implica qualquer uma das outras, use simplesmente silogismo hipotético repetidamente. **39.** Daremos uma demonstração por contradição. Suponha que $a_1, a_2, \ldots, a_n$ sejam todos menores do que A, em que A é a média desses números. Então, $a_1 + a_2 + \cdots + a_n < nA$. Dividir ambos os lados por n mostra que $A + (a_1 + a_2 + \cdots + a_n)/n < A$, que é uma contradição. **41.** Mostraremos que as quatro afirmações são equivalentes indicando que (*i*) implica (*ii*), (*ii*) implica (*iii*), (*iii*) implica (*iv*) e (*iv*) implica (*i*). Primeiro, suponha que n seja par. Então, $n + 2k$ para algum inteiro k. Assim, $n + 1 = 2k + 1$, de modo que $n + 1$ é ímpar. Isso mostra que (*i*) implica (*ii*). A seguir, suponha que $n + 1$ seja ímpar, de modo que $n + 1 = 2k + 1$ para algum inteiro k. Então, $3n + 1 = 2n + (n + 1) = 2(n + k) + 1$, o que mostra que $3n + 1$ é ímpar, indicando que (*ii*) implica (*iii*). A seguir, suponha que $3n + 1$ seja ímpar, de modo que $3n + 1 = 2k + 1$ para algum inteiro k. Assim, $3n = (2k + 1) - 1 = 2k$, de modo que $3n$ é par. Isso mostra que (*iii*) implica (*iv*). Finalmente, suponha que n não seja par. Então, n é ímpar, de modo que $n = 2k + 1$ para algum inteiro k. Logo, $3n = 3(2k + 1) = 6k + 3 = 2(3k + 1) + 1$, e $3n$ é ímpar. Isso completa a demonstração por contraposição de que (*iv*) implica (*i*).

Seção 1.7

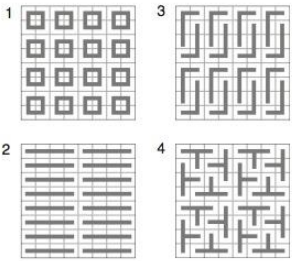
1. $1^2 + 1 = 2 \geq 2 = 2^1$; $2^2 + 1 + 5 \geq 4 = 2^2$; $3^2 + 1 = 10 \geq 8 = 2^3$; $4^2 + 1 = 17 \geq 16 = 2^4$ **3.** Se $x \leq y$, então $\max(x, y) + \min(x, y) = y + x = x + y$. Se $x \geq y$, então $\max(x, y) + \min(x, y) = x + y$. Como estes são os dois únicos casos, a igualdade sempre é válida. **5.** Existem quatro casos. *Caso 1:* $x \geq 0$ e $y \geq 0$. Então, $|x| + |y| = x + y = |x + y|$. *Caso 2:* $x < 0$ e $y < 0$. Então, $|x| + |y| = -x + (-y) = -(x + y) = |x + y|$, pois $x + y < 0$. *Caso 3:* $x \geq 0$ e $y < 0$. Então, $|x| + |y| = x + (-y)$. Se $x \geq -y$, então $|x + y| = x + y$. Mas, como $y < 0$, $-y > y$, de modo que $|x| + |y| = x + (-y) > x + y = |x + y|$. Se $x < -y$, então $|x + y| = -(x + y) = -x + (-y)$. Mas, como $x \geq 0$, $x \geq -x$, de modo que $|x| + |y| = x + (-y) \geq -x + (-y) = |x + y|$. *Caso 4:* $x < 0$ e $y \geq 0$. Idêntico ao Caso 3, com os papéis de x e y trocados. **7.** 10.001, 10.002, ..., 10.100 não são quadrados perfeitos, pois $100^2 = 10.000$ e $101^2 = 10.201$; construtivo. **9.** $8 = 2^3$ e $9 = 3^2$ **11.** Sejam $x = 2$ e $y = \sqrt{2}$. Se $x^y = 2^{\sqrt{2}}$ for irracional, teremos acabado. Se não, então sejam $x = 2^{\sqrt{2}}$ e $y = \sqrt{2}/4$. Então $x^y = (2^{\sqrt{2}})^{\sqrt{2}/4} = 2^{\sqrt{2}\cdot(\sqrt{2})/4} = 2^{1/2} = \sqrt{2}$. **13. a)** Esta afirmação garante a existência de x com determinada propriedade. Se fizermos $y = x$, então vemos que $P(x)$ é verdadeira. Se y for qualquer coisa diferente de x, então $P(x)$ não é verdadeira. Logo, x é o único elemento que torna P verdadeira. **b)** A primeira cláusula aqui diz que existe um elemento que torna P verdadeira. A segunda cláusula diz que sempre que dois elementos tornarem P verdadeira, eles devem, na realidade, ser o mesmo elemento. Juntos, elas dizem que P é satisfeita por exatamente um elemento. **c)** Esta afirmação garante a existência de um x que torna P verdadeira e tem a propriedade adicional que sempre que encontremos um elemento que torna P verdadeira, este elemento é x. Em outras palavras, x é o único elemento que torna P verdadeira. **15.** A equação $|a - c| = |b - c|$ é equivalente à disjunção de duas equações: $a - c = b - c$ ou $a - c = -b + c$. A primeira destas é equivalente a $a = b$, o que contraria a hipótese feita neste problema, de modo que a equação original é equivalente a $a - c = -b + c$. Somando $b + c$ a ambos os lados e dividindo por 2, vemos que esta equação é equivalente a $c = (a + b)/2$. Logo, existe uma única solução. Além disso, este c é um inteiro, pois a soma de inteiros ímpares a e b é par. **17.** Nos foi pedido que isolássemos k em $n = (k - 2) + (k + 3)$. Usando as regras usuais, reversíveis, da álgebra, vemos que esta equação é equivalente a $k = (n - 1)/2$. Em outras palavras, este é o único valor de k que torna nossa equação verdadeira. Como n é ímpar, $n - 1$ é par, de modo que k é um inteiro. **19.** Se x for ele próprio um inteiro, então podemos tomar $n = x$ e $\in = 0$. Nenhuma outra solução é possível neste caso, pois, se o inteiro n for maior que x, então n é pelo menos $x + 1$, o que tornaria $\in \geq 1$. Se x não for um inteiro, então, arredonde-o para o próximo inteiro, e chame este inteiro de n. Seja $\in = n - x$. Claramente, $0 \leq \in < 1$; este é o único $\in$ que funcionará com este n, e n não pode ser maior, pois $\in$ é restrito a ser menor que 1. **21.** A média harmônica de dois números reais positivos distintos x e y é sempre menor que a média geométrica. Para demonstrar que $2xy/(x + y) < \sqrt{xy}$, multiplique ambos os lados por $(x + y)/(2\sqrt{xy})$ para obter a desigualdade equivalente $\sqrt{xy} < (x + y)/2$, que foi demonstrada no Exemplo 14. **23.** A paridade (par ou ímpar) da soma de números escritos na lousa nunca muda, pois $j + k$ e $|j - k|$ têm a mesma paridade (e, em cada passo, reduzimos a soma por $j + k$, mas a aumentamos por $|j - k|$). Portanto, o inteiro no final do processo deve ter a mesma paridade de $1 + 2 + \cdots + (2n) = n(2n + 1)$, que é ímpar, pois n é ímpar. **25.** Sem perda de generalidade, podemos supor que n é não negativo, pois a quarta potência de um inteiro e a quarta potência de seu oposto são a mesma coisa. Dividimos um inteiro positivo arbitrário n por 10, obtendo um quociente k e um resto l, o que implica que $n = 10k + l$, e l é um inteiro entre 0 e 9, inclusive. Então, podemos calcular n^4 em cada um destes 10 casos. Obtemos os seguintes valores, em que X é algum inteiro que é um múltiplo de 10, cujo valor exato não nos interessa. $(10k + 0)^4 = 10{,}000k^4 + 0$, $(10k + 1)^4 = 10{,}000k^4 + X \cdot k^3 + X \cdot k^2 + X \cdot k + 1$, $(10k + 2)^4 = 10{,}000k^4 + X \cdot k^3 + X \cdot k^2 + X \cdot k + 16$, $(10k + 3)^4 = 10{,}000k^4 + X \cdot k^3 + X \cdot k^2 + X \cdot k + 81$, $(10k + 4)^4 = 10{,}000k^4 + X \cdot k^3 + X \cdot k^2 + X \cdot k + 256$, $(10k + 5)^4 = 10{,}000k^4 + X \cdot k^3 + X \cdot k^2 + X \cdot k + 625$, $(10k + 6)^4 = 10{,}000k^4 + X \cdot k^3 + X \cdot k^2 + X \cdot k + 1296$, $(10k + 7)^4 = 10{,}000k^4 + X \cdot k^3 + X \cdot k^2 + X \cdot k + 2401$, $(10k + 8)^4 = 10{,}000k^4 + X \cdot k^3 + X \cdot k^2 + X \cdot k + 4096$, $(10k + 9)^4 = 10{,}000k^4 + X \cdot k^3 + X \cdot k^2 + X \cdot k + 6561$. Como cada coeficiente indicado por X é um múltiplo de 10, o termo correspondente não tem efeito nos primeiros algarismos da resposta. Portanto, os primeiros algarismos são 0, 1, 6, 1, 6, 5, 6, 1, 6, 1, respectivamente, de modo que ele é sempre 0, 1, 5 ou

6. **27.** Como $n^3 > 100$ para todo $n > 4$, precisamos apenas observar que $n = 1$, $n = 2$, $n = 3$ e $n = 4$ não satisfazem $n^2 + n^3 = 100$. **29.** Como $5^4 = 625$, ambos, x e y, devem ser menores que 5. Então, $x^4 + y^4 \le 4^4 + 4^4 = 512 < 625$. **31.** Se não for verdade que $a \le \sqrt[3]{n}$, $b \le \sqrt[3]{n}$ ou $c \le \sqrt[3]{n}$, então $a > \sqrt[3]{n}$, $b > \sqrt[3]{n}$ e $c > \sqrt[3]{n}$. Multiplicando estas desigualdades de números positivos, obtemos $abc < (\sqrt[3]{n})^3 = n$, o que implica a negação de nossa hipótese que $n = abc$ **33.** Encontrando um denominador comum, podemos supor que os números racionais dados são a/b e c/b, em que b é um inteiro positivo e a e c são inteiros com $a < c$. Em particular, $(a + 1)/b \le c/b$. Logo, $x = (a + \frac{1}{2}\sqrt{2})/b$ está entre os dois números racionais dados, pois $0 < \sqrt{2} < 2$. Além disso, x é irracional, pois se x fosse racional, então $2(bx - a) = \sqrt{2}$ também seria, o que viola o Exemplo 10 da Seção 1.6. **35. a)** Sem perda de generalidade, podemos supor que a seqüência x já está ordenada na ordem não decrescente, pois podemos rerrotular os índices. Existe apenas um número finito de ordens possíveis para a seqüência y, de modo que se pudermos mostrar que podemos aumentar a soma (ou pelo menos mantê-la a mesma) sempre que encontrarmos y_i e y_j que estão fora de ordem (isto é, $i < j$, mas $y_i > y_j$) trocando-os, então teremos mostrado que a soma é maior quando a seqüência y estiver na ordem não decrescente. De fato, se fizermos a troca, então teremos adicionado $x_i y_j + x_j y_i$ à soma e subtraído $x_i y_i + x_j y_j$. O efeito resultante é o de adicionar $x_i y_j + x_j y_i - x_i y_i - x_j y_j = (x_j - x_i)(y_i - y_j)$, que é não negativo, por nossas hipóteses de ordenação. **b)** Análogo à parte (a). **37. a)** $6 \to 3 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$ **b)** $7 \to 22 \to 11 \to 34 \to 17 \to 52 \to 26 \to 13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$ **c)** $17 \to 52 \to 26 \to 13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$ **d)** $21 \to 64 \to 32 \to 16 \to 8 \to 4 \to 2 \to 1$ **39.** Sem perda de generalidade, suponha que os cantos superiores direito e esquerdo do tabuleiro sejam removidos. Coloque três dominós horizontalmente para encher a parte restante da primeira linha e encha cada uma das outras sete linhas com quatro dominós horizontais. **41.** Como existe um número par de quadrados no total, ou há um número par de quadrados em cada linha ou existe um número par de quadrados em cada coluna. No primeiro caso, ladrilhe o tabuleiro da maneira óbvia, colocando dominós horizontalmente, e no último caso, ladrilhe o tabuleiro da maneira óbvia, colocando dominós verticalmente. **43.** Podemos girar o tabuleiro se necessário para fazer os quadrados removidos serem o 1 e o 16. O quadrado 2 deve ser coberto por um dominó. Se aquele dominó for colocado para cobrir os quadrados 2 e 6, então a seguinte colocação de dominós é forçada em seqüência: 5-9, 13-14 e 10-11, em cujo ponto não há maneira de cobrir o quadrado 15. Caso contrário, o quadrado 2 deve ser coberto por um dominó colocado em 2-3. Então, a seguinte colocação de dominós é forçada: 4-8, 11-12, 6-7, 5-9 e 10-14, e novamente não há maneira de cobrir o quadrado 15. **45.** Remova dois quadrados pretos adjacentes a um canto branco, e remova dois quadrados brancos distintos dos cantos. Então, nenhum dominó pode cobrir aquele canto branco.

47. a)



b) A figura mostra ladrilhamentos para os primeiros quatro padrões.



Para mostrar que o padrão 5 não pode ladrilhar o tabuleiro de xadrez, numere os quadrados de 1 a 64, uma linha de cada vez, a partir de cima, da esquerda para a direita em cada linha. Assim, o quadrado 1 é o canto superior esquerdo, e o quadrado 64 é o canto inferior direito. Suponha que tenhamos um ladrilhamento. Por simetria e sem perda de generalidade, podemos supor que a peça está posicionada no canto superior esquerdo, cobrindo os quadrados 1, 2, 10 e 11. Isso força uma peça a ser adjacente a esta à direita, cobrindo os quadrados 3, 4, 12 e 13. Continue desta maneira e seremos forçados a ter uma peça cobrindo os quadrados 6, 7, 15 e 16. Isso torna impossível recobrir o quadrado 8. Assim, não é possível nenhum ladrilhamento.

Exercícios Complementares

1. a) $q \to p$ **b)** $q \wedge p$ **c)** $\neg q \vee \neg p$ **d)** $q \leftrightarrow p$ **3. a)** A proposição não pode ser falsa, a menos que $\neg p$ seja falsa, de modo que p é verdadeira. Se p é verdadeira e q é verdadeira, então $\neg q \wedge (p \to q)$ é falsa, de modo que a afirmação condicional é verdadeira. Se p for verdadeira e q for falsa, então $p \to q$ é falsa, de modo que $\neg q \wedge (p \to q)$ é falsa e a afirmação condicional é verdadeira. **b)** A proposição não pode ser falsa, a menos que q seja falsa. Se q for falsa e p for verdadeira, então $(p \vee q) \wedge \neg p$ é falsa, e a afirmação condicional é verdadeira. Se q for falsa e p for falsa, então $(p \vee q) \wedge \neg p$ é falsa, e a afirmação condicional é verdadeira. **5.** $\neg q \to \neg p$; $p \to q$; $\neg p \to \neg q$ **7.** $(p \wedge q \wedge r \wedge \neg s) \vee (p \wedge q \wedge \neg r \wedge s) \vee (p \wedge \neg q \wedge r \wedge s) \vee (\neg p \wedge q \wedge r \wedge s)$ **9.** Traduzindo estas sentenças em símbolos, usando as letras óbvias, temos $\neg t \to \neg g$, $\neg g \to \neg q$, $r \to q$ e $\neg t \wedge r$. Suponha que as sentenças sejam consistentes. A quarta sentença nos diz que $\neg t$ deve ser verdadeira. Portanto, *por modus ponens* com a primeira sentença, sabemos que $\neg g$ é verdadeira, e assim (a partir da segunda sentença), que $\neg q$

é verdadeira. Além disso, a quarta sentença nos diz que r deve ser verdadeira, e assim, novamente, *modus ponens* (terceira sentença) torna q verdadeira. Isso é uma contradição: $q \wedge \neg q$. Logo, as sentenças são inconsistentes. **11.** Bela. **13.** As premissas não podem ser ambas verdadeiras, pois elas são contraditórias. Portanto, é (trivialmente) verdade que sempre que todas as premissas sejam verdadeiras, a conclusão também é verdadeira, o que por definição tornaria isso um argumento válido. Como as premissas não são ambas verdadeiras, não podemos concluir que a conclusão é verdadeira. **15. a)** F **b)** V **c)** F **d)** V **e)** F **f)** V **17.** São possíveis muitas respostas. Um exemplo são os senadores dos Estados Unidos. **9.** $\forall x \exists y \exists z\ (y \neq z \wedge \forall w(P(w, x) \leftrightarrow (w = y \vee w = z)))$ **21. a)** $\neg \exists x\ P(x)$ **b)** $\exists x(P(x) \wedge \forall y(P(y) \rightarrow y = x))$ **c)** $\exists x_1 \exists x_2(P(x_1) \wedge P(x_2) \wedge x_1 \neq x_2 \wedge \forall y\,(P(y) \rightarrow (y = x_1 \vee y = x_2)))$ **d)** $\exists x_1 \exists x_2 \exists x_3(P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3 \wedge \forall y(P(y) \rightarrow (y = x_1 \vee y = x_2 \vee y = x_3)))$ **23.** Suponha que $\exists x(P(x) \rightarrow Q(x))$ seja verdadeira. Então, ou $Q(x_0)$ é verdadeira para algum x_0, em cujo caso $\forall x\ P(x) \rightarrow \exists x\ Q(x)$ é verdadeira; ou $P(x_0)$ é falsa para algum x_0, em cujo caso $\forall x\ P(x) \rightarrow \exists x\ Q(x)$ é verdadeira. Reciprocamente, suponha que $\exists x(P(x) \rightarrow Q(x))$ seja falsa. Isso significa que $\forall x(P(x) \wedge \neg Q(x))$ é verdadeira, o que implica que $\forall x\ P(x)$ e $\forall x(\neg Q(x))$. Esta última proposição é equivalente a $\neg \exists x\ Q(x)$. Logo, $\forall x\ P(x) \rightarrow \exists x\ Q(x)$ é falsa. **25.** Não. **27.** $\forall x\ \forall z\ \exists y\ T(x, y, z)$, em que $T(x, y, z)$ é a afirmação que o estudante x fez algum curso y no departamento z, em que os domínios são o conjunto de estudantes na classe, o conjunto de cursos nesta universidade e o conjunto de departamentos na escola de ciências matemáticas. **29.** $\exists! x \exists! y\ T(x, y)$ e $\exists x \forall z((\exists y \forall w(T(z, w) \leftrightarrow w = y)) \leftrightarrow z = x)$, em que $T(x, y)$ significa que o estudante x teve um curso y e o domínio são todos os estudantes na classe. **31.** $P(a) \rightarrow Q(a)$ e $Q(a) \rightarrow R(a)$ por instanciação universal; então, $\neg Q(a)$ por *modus tollens* e $\neg P(a)$ por *modus tollens* **33.** Damos uma demonstração por contraposição e mostramos que, se $\sqrt{x}$ for racional, então x é racional, supondo que $x \geq 0$. Suponha que $\sqrt{x} = p/q$ seja racional, $q \neq 0$. Então, $x = (\sqrt{x})^2 = p^2/q^2$ também é racional (q^2 novamente é não nulo). **35.** Podemos dar uma demonstração construtiva tomando $m = 10^{500} + 1$. Então, $m = (10^{500} + 1)^2 > (10^{500})^2 = 10^{1000}$. **37.** 23 não pode ser escrito como a soma de oito cubos. **39.** 223 não pode ser escrito como a soma de 36 quintas potências.

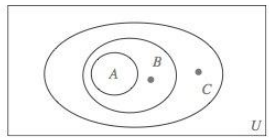
CAPÍTULO 2

Seção 2.1

1. a) $\{-1, 1\}$ **b)** $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ **c)** $\{0, 1, 4, 9, 16, 25, 36, 49, 64, 81\}$ **d)** $\emptyset$ **3. a)** Sim **b)** Não **c)** Não **5. a)** Sim **b)** Não **c)** Sim **d)** Não **e)** Não **f)** Não **7. a)** Falsa **b)** Falsa **c)** Falsa **d)** Verdadeira **e)** Falsa **f)** Falsa **g)** Verdadeira **9. a)** Verdadeira **b)** Verdadeira **c)** Falsa **d)** Verdadeira **e)** Verdadeira **f)** Falsa

11.

13. Os pontos em certas regiões indicam que aquelas regiões não são vazias.



15. Suponha que $x \in A$. Como $A \subseteq B$, isso implica que $x \in B$. Como $B \subseteq C$, vemos que $x \in C$. Como $x \in A$ implica que $x \in C$, segue que $A \subseteq C$. **17. a)** 1 **b)** 1 **c)** 2 **d)** 3 **19. a)** $\{\emptyset, \{a\}\}$ **b)** $\{\emptyset, \{a\}, \{b\}, \{a, b\}\}$ **c)** $\{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$ **21. a)** 8 **b)** 16 **c)** 2 **23. a)** $\{(a, y), (b, y), (c, y), (d, y), (a, z), (b, z), (c, z), (d, z)\}$ **b)** $\{(y, a), (y, b), (y, c), (y, d), (z, a), (z, b), (z, c), (z, d)\}$ **25.** O conjunto de trios (a, b, c), em que a é uma empresa aérea e b e c são cidades. **27.** $\emptyset \times A = \{(x, y) \mid x \in \emptyset \text{ e } y \in A\} = \emptyset = \{(x, y) \mid x \in A \text{ e } y \in \emptyset\} = A \times \emptyset$ **29.** mn **31.** Os elementos de $A \times B \times C$ consistem nas triplas (a, b, c), em que $a \in A$, $b \in B$ e $c \in C$, enquanto os elementos de $(A \times B) \times C$ se parecem com $((a, b), c)$ — pares ordenados, a primeira coordenada dos quais é de novo um par ordenado. **33. a)** O quadrado de um número real nunca é -1. Verdadeira. **b)** Existe um inteiro cujo quadrado é 2. Falsa. **c)** O quadrado de todo inteiro é positivo. Falsa. **d)** Existe um número real igual a seu próprio quadrado. Verdadeira **35. a)** $\{-1, 0, 1\}$ **b)** $\mathbf{Z} - \{0, 1\}$ **c)** $\emptyset$ **37.** Devemos mostrar que $\{\{a\}, \{a, b\}\} = \{\{c\}, \{c, d\}\}$ se e somente se $a = c$ e $b = d$. A parte "se" é imediata. Assim, suponha que estes dois conjuntos sejam iguais. Primeiro, considere o caso em que $a \neq b$. Então $\{\{a\}, \{a, b\}\}$ contém exatamente dois elementos, um dos quais contém um elemento. Logo, $\{\{c\}, \{c, d\}\}$ deve ter a mesma propriedade, de modo que $c \neq d$ e $\{c\}$ é o elemento que contém exatamente um elemento. Logo, $\{a\} = \{c\}$, o que implica que $a = c$. Além disso, os conjuntos de dois elementos $\{a, b\}$ e $\{c, d\}$ devem ser iguais. Como $a = c$ e $a \neq b$, segue que $b = d$. Segundo, suponha que $a = b$. Então, $\{\{a\}, \{a, b\}\} = \{\{a\}\}$, um conjunto com um elemento. Logo, $\{\{c\}, \{c, d\}\}$ tem apenas um elemento, o que pode ocorrer apenas quando $c = d$, e o conjunto for $\{\{c\}\}$. Segue então que $a = c$ e $b = d$. **39.** Seja $S = \{a_1, a_2, \ldots, a_n\}$. Represente cada subconjunto de S por uma seqüência de bits de comprimento n, em que o i-ésimo bit é 1 se e somente se $a_i \in S$. Para gerar todos os subconjuntos de S, liste todas as 2^n seqüências de bits de comprimento n (por exemplo, em ordem crescente), e escreva os subconjuntos correspondentes.

Seção 2.2

1. a) O conjunto dos estudantes que moram a um quilômetro de distância da faculdade e que vão a pé para as aulas. **b)** O conjunto dos estudantes que moram a um quilômetro de distância da faculdade ou que vão a pé para as aulas (ou ambas as coisas). **c)** O conjunto dos estudantes que moram a um quilômetro de distância da faculdade, mas que não vão a pé para as aulas. **d)** O conjunto dos estudantes que vão a pé para as aulas, mas moram a mais de um quilômetro de distância da faculdade **3. a)** $\{0, 1, 2, 3, 4, 5, 6\}$ **b)** $\{3\}$ **c)** $\{1, 2, 4, 5\}$ **d)** $\{0, 6\}$ **5.** $\overline{\overline{A}} = \{x \mid \neg(x \in \overline{A})\} = \{x \mid \neg(\neg x \in A)\} = \{x \mid x \in A\} = A$ **7. a)** $A \cup U = \{x \mid x \in A \vee x \in U\} = \{x \mid x \in A \vee \mathbf{T}\} = \{x \mid \mathbf{T}\} = U$ **b)** $A \cap \emptyset = \{x \mid x \in A \wedge x \in \emptyset\} = \{x \mid x \in A \wedge \mathbf{F}\} = \{x \mid \mathbf{F}\} = \emptyset$ **9. a)** $A \cup \overline{A} = \{x \mid x \in A \vee x \notin A\} = U$ **b)** $A \cap \overline{A} = \{x \mid x \in A \wedge x \notin A\} = \emptyset$ **11. a)** $A \cup B = \{x \mid x \in A \vee x \in B\} = \{x \mid x \in B \vee x \in A\} = B \cup A$ **b)** $A \cap B = \{x \mid x \in A \wedge x \in B\} = \{x \mid x \in B \wedge x \in A\} = B \cap A$ **13.** Suponha que $x \in A \cap (A \cup B)$. Então, $x \in A$ e $x \in A \cup B$ pela definição de intersecção. Como $x \in A$, demonstramos que o lado esquerdo é um subconjunto do lado direito. Reciprocamente, seja $x \in A$. Então, pela definição de união, $x \in A \cup B$ também. Portanto, $x \in A \cap (A \cup B)$ pela definição de intersecção, de modo que o lado direito é um subconjunto do lado esquerdo. **15. a)** $x \in \overline{(A \cup B)} \equiv x \notin (A \cup B) \equiv \neg(x \in A \vee x \in B) \equiv \neg(x \in A) \wedge \neg(x \in B) \equiv x \notin A \wedge x \notin B \equiv x \in \overline{A} \wedge x \in \overline{B} \equiv x \in \overline{A} \cap \overline{B}$

b)

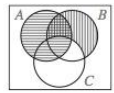
A	B	$A \cup B$	$\overline{(A \cup B)}$	$\overline{A}$	$\overline{B}$	$\overline{A} \cap \overline{B}$
1	1	1	0	0	0	0
1	0	1	0	0	1	0
0	1	1	0	1	0	0
0	0	0	1	1	1	1

17. a) $x \in \overline{A \cap B \cap C} \equiv x \notin A \cap B \cap C \equiv x \notin A \vee x \notin B \vee x \notin C \equiv x \in \overline{A} \vee x \in \overline{B} \vee x \in \overline{C} \equiv x \in \overline{A} \cup \overline{B} \cup \overline{C}$

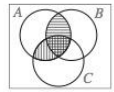
b)

A	B	C	$A \cap B \cap C$	$\overline{(A \cap B \cap C)}$	$\overline{A}$	$\overline{B}$	$\overline{C}$	$\overline{A} \cup \overline{B} \cup \overline{C}$
1	1	1	1	0	0	0	0	0
1	1	0	0	1	0	0	1	1
1	0	1	0	1	0	1	0	1
1	0	0	0	1	0	1	1	1
0	1	1	0	1	1	0	0	1
0	1	0	0	1	1	0	1	1
0	0	1	0	1	1	1	0	1
0	0	0	0	1	1	1	1	1

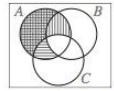
19. Ambos os lados são iguais a $\{x \mid x \in A \wedge x \notin B\}$. **21.** $x \in A \cup (B \cup C) \equiv (x \in A) \vee (x \in (B \cup C)) \equiv (x \in A) \vee (x \in B \vee x \in C) \equiv (x \in A \vee x \in B) \vee (x \in C) \equiv x \in (A \cup B) \cup C$ **23.** $x \in A \cup (B \cap C) \equiv (x \in A) \vee (x \in (B \cap C)) \equiv (x \in \mathrm{A}) \vee (x \in B \wedge x \in C) \equiv (x \in A \vee x \in B) \wedge (x \in A \vee x \in C) \equiv x \in (A \cup B) \cap (A \cup C)$ **25. a)** $\{4, 6\}$ **b)** $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ **c)** $\{4,5,6,8,10\}$ **d)** $\{0,2,4,5,6,7,8,9,10\}$ **27. a)** A parte duplamente sombreada é o conjunto desejado.



b) O conjunto desejado é toda a parte sombreada.



c) O conjunto desejado é toda a parte sombreada.

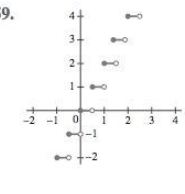


29. a) $B \subseteq A$ **b)** $A \subseteq B$ **c)** $A \cap B = \emptyset$ **d)** Nada, pois isto é sempre verdade. **e)** $A = B$ **31.** $A \subseteq B \equiv \forall x\,(x \in A \rightarrow x \in B) \equiv \forall x\,(x \notin B \rightarrow x \notin A) \equiv \forall x\,(x \in \overline{B} \rightarrow x \in \overline{A}) \equiv \overline{B} \subseteq \overline{A}$ **33.** O conjunto de estudantes de ciência da computação, mas não de matemática ou de matemática mas não de ciência da computação. **35.** Um elemento está em $(A \cup B) - (A \cap B)$ se estiver na união de A e B, mas não na intersecção de A e B, o que significa que ele está ou em A ou em B, mas não em ambos, A e B. Isto é exatamente o que significa um elemento pertencer a $A \oplus B$. **37. a)** $A \oplus A = (A - A) \cup (A - A) = \emptyset \cup \emptyset = \emptyset$ **b)** $A \oplus \emptyset = (A - \emptyset) \cup (\emptyset - A) = A \cup \emptyset = A$ **c)** $A \oplus U = (A - U) \cup (U - A) = \emptyset \cup \overline{A} = \overline{A}$ **d)** $A \oplus \overline{A} = (A - \overline{A}) \cup (\overline{A} - A) = A \cup \overline{A} = U$ **39.** $B = \emptyset$ **41.** Sim **43.** Sim **45. a)** $\{1, 2, 3, \ldots, n\}$ **b)** $\{1\}$ **47. a)** A_n **b)** $\{0, 1\}$ **49. a)** $\mathbf{Z}, \{-1, 0, 1\}$ **b)** $\mathbf{Z} - \{0\}, \emptyset$ **c)** $\mathbf{R}$, $[-1, 1]$ **d)** $[1, \infty), \emptyset$ **51. a)** $\{1, 2, 3, 4, 7, 8, 9, 10\}$ **b)** $\{2, 4, 5, 6, 7\}$ **c)** $\{1, 10\}$ **53.** O bit na i-ésima posição da cadeia de bits de dois conjuntos diferentes é 1 se o i-ésimo bit da primeira seqüência for 1 e o i-ésimo bit da segunda seqüência for 0, e é 0 caso contrário. **55. a)** 11 1110 0000 0000 0000 0000 0000 $\vee$ 01 1100 1000 0000 0100 0101 0000 = 11 1110 1000 0000 0100 0101 0000, representando $\{a, b, c, d, e, g, p, t, v\}$ **b)** 11 1110 0000 0000 0000 0000 0000 $\wedge$ 01 1100 1000 0000 0100 0101 0000 = 01 1100 0000 0000 0000 0000 0000, representando $\{b, c, d\}$ **c)** (11 1110 0000 0000 0000 0000 0000 $\vee$ 00 0110 0110 0001 1000 0110 0110) $\wedge$ (01 1100 1000 0000 0100 0101 0000 $\vee$ 00 1010 0010 0000 1000 0010 0111) = 11 1110 0110 0001 1000 0110 0110 $\wedge$ 01 1110 1010 0000 1100 0111 0111 = 01 1110 0010 0000 1000 0110 0110, representando $\{b, c, d, e, i, o, t, u, x, y\}$ **d)** 11 1110 0000 0000 0000 0000 0000 $\vee$ 01 1100 1000 0000 0100 0101 0000 $\vee$ 00 1010 0010 0000 1000 0010 0111 $\vee$ 00 0110 0110 0001 1000 0110 0110 = 11 1110 1110 0001 1100 0111 0111, representando $\{a, b, c, d, e, g, h, i, n, o, p, t, u, v, x, y, z\}$ **57. a)** $\{1, 2, 3, \{1, 2, 3\}\}$ **b)** $\{\emptyset\}$ **c)** $\{\emptyset, \{\emptyset\}\}$ **d)** $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$ **59. a)** $\{3 \cdot a, 3 \cdot b, 1 \cdot c, 4 \cdot d\}$ **b)** $\{2 \cdot a, 2 \cdot b\}$ **c)** $\{1 \cdot a, 1 \cdot c\}$ **d)** $\{1 \cdot b, 4 \cdot d\}$ **e)** $\{5 \cdot a, 5 \cdot b, 1 \cdot c, 4 \cdot d\}$ **61.** $\overline{F} = \{0{,}4$ Alice, 0,1 Brian, 0,6 Fred, 0,9 Oscar, 0,5 Rita$\}$, $\overline{R} = \{0{,}6$ Alice, 0,2 Brian, 0,8 Fred, 0,1 Oscar, 0,3 Rita$\}$ **63.** $F \cap R = \{0{,}4$ Alice, 0.8 Brian, 0,2 Fred, 0,1 Oscar, 0.5 Rita$\}$

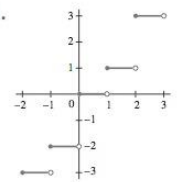
Seção 2.3

1. a) $f(0)$ não está definida. **b)** $f(x)$ não está definida para $x < 0$. **c)** $f(x)$ não está bem definida, pois existem dois valores distintos associados a cada x. **3. a)** Não é uma função. **b)** É uma função. **c)** Não é uma função. **5. a)** Domínio, o conjunto das cadeias de bits; imagem, o conjunto dos inteiros. **b)** Domínio, o conjunto das cadeias de bits; imagem, o conjunto dos inteiros pares não negativos. **c)** Domínio, o conjunto das cadeias de bits; imagem, o conjunto dos inteiros não negativos que não excedem. **7. d)** Domínio, o conjunto dos inteiros positivos; imagem, o conjunto dos quadrados de inteiros positivos $= \{1, 4, 9, 16, \ldots\}$ **7. a)** Domínio, $\mathbf{Z}^+ \times \mathbf{Z}^+$; imagem, $\mathbf{Z}^+$. **b)** Domínio, $\mathbf{Z}^+$; imagem, $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. **c)** Domínio, o conjunto das cadeias de bits; imagem, **N**. **d)** Domínio, o conjunto das cadeias de bits; imagem, **N**. **9. a)** 1 **b)** 0 **c)** 0 **d)** -1 **e)** 3 **f)** -1 **g)** 2 **h)** 1 **11.** Apenas a função na parte (a). **13.** Apenas as funções nas partes (a) e (d). **15. a)** Sobrejetiva. **b)** Não é sobrejetiva **c)** Sobrejetiva. **d)** Não é sobrejetiva. **e)** Sobrejetiva. **17. a)** A função $f(x)$ com $f(x) = 3x + 1$ quando $x \geq 0$ e $f(x) = -3x + 2$ quando $x < 0$ **b)** $f(x) = |x| + 1$ **c)** A função $f(x)$ com $f(x) = 2x + 1$ quando $x \geq 0$ e $f(x) = -2x$ quando $x < 0$ **d)** $f(x) = x^2 + 1$ **19. a)** Sim. **b)** Não. **c)** Sim. **d)** Não. **21.** Suponha que f seja estritamente decrescente. Isto significa que $f(x) > f(y)$ sempre que $x < y$. Para mostrar que g é estritamente crescente, suponha que $x < y$. Então, $g(x) = 1/f(x) < 1/f(y) = g(y)$. Reciprocamente, suponha que g seja estritamente crescente. Isto significa que $g(x) < g(y)$ sempre que $x < y$. Para mostrar que f é estritamente decrescente, suponha que $x < y$. Então, $f(x) = 1/g(x) > 1/g(y) = f(y)$. **23.** São possíveis muitas respostas. Um exemplo é $f(x) = 17$. **25.** A função não é injetora, de modo que não é inversível. No domínio restrito, a função é a função identidade nos números reais não negativos, $f(x) = x$, de modo que ela é sua própria inversa. **27. a)** $f(S) = \{0, 1, 3\}$ **b)** $f(S) = \{0, 1, 3, 5, 8\}$ **c)** $f(S) = \{0, 8, 16, 40\}$ **d)** $f(S) = \{1, 12, 33, 65\}$ **29. a)** Sejam x e y elementos distintos de A. Como g é injetora, $g(x)$ e $g(y)$ são elementos distintos de B. Como f é injetora, $f(g(x)) = (f \circ g)(x)$ e $f(g(y)) = (f \circ g)(y)$ são elementos distintos de C. Portanto, $f \circ g$ é injetora. **b)** Seja $y \in C$. Como f é sobrejetiva, $y = f(b)$ para algum $b \in B$. Agora, como g é sobrejetiva, $b = g(x)$ para algum $x \in A$. Portanto, $y = f(b) = f(g(x)) = (f \circ g)(x)$. Segue que $f \circ g$ é sobrejetiva. **31.** Não. Por exemplo, suponha que $A = \{a\}$, $B = \{b, c\}$ e $C = \{d\}$. Sejam $g(a) = b$, $f(b) = d$ e $f(c) = d$. Então, f e $f \circ g$ são sobrejetivas, mas g não é. **33.** $(f+g)(x) = x^2 + x + 3$, $(fg)(x) = x^3 + 2x^2 + x + 2$ **35.** f é injetora, pois $f(x_1) = f(x_2) \rightarrow ax_1 + b = ax_2 + b \rightarrow ax_1 = ax_2 \rightarrow x_1 = x_2$. f é sobrejetiva, pois $f((y-b)/a) = y$. $f^{-1}(y) = (y-b)/a$. **37.** Sejam $f(1) = a$, $f(2) = a$. Sejam $S = \{1\}$ e $T = \{2\}$. Então, $f(S \cap T) = f(\emptyset) = \emptyset$, mas $f(S) \cap f(T) = \{a\} \cap \{a\} = \{a\}$. **39. a)** $\{x \mid 0 \leq x < 1\}$ **b)** $\{x \mid -1 \leq x < 2\}$ **c)** $\emptyset$ **41.** $f^{-1}(\overline{S}) = \{x \in A \mid f(x) \notin S\} = \overline{\{x \in A \mid f(x) \in S\}} = \overline{f^{-1}(S)}$ **43.** Seja $x = \lfloor x \rfloor + \epsilon$, em que ϵ é um número real com $0 \leq \epsilon < 1$. Se $\epsilon < \frac{1}{2}$, então $\lfloor x \rfloor - 1 < x - \frac{1}{2} < \lfloor x \rfloor$, de modo que $\lceil x - \frac{1}{2} \rceil = \lfloor x \rfloor$ e este é o inteiro mais próximo de x. Se $\epsilon > \frac{1}{2}$ então $\lfloor x \rfloor < x - \frac{1}{2} < \lfloor x \rfloor + 1$, de modo que $\lceil x - \frac{1}{2} \rceil = \lfloor x \rfloor + 1$ e este é o inteiro mais próximo de x. Se $\epsilon = \frac{1}{2}$, então $\lceil x - \frac{1}{2} \rceil = \lfloor x \rfloor$, que é o menor dos dois inteiros que cercam x e estão à mesma distância de x. **45.** Escreva o número real x como $\lfloor x \rfloor + \epsilon$, em que ϵ é um número real com $0 \leq \epsilon < 1$. Como $\epsilon = x - \lfloor x \rfloor$, segue que $0 \leq -\lfloor x \rfloor < 1$. As duas primeiras desigualdades, $x - 1 < \lfloor x \rfloor$ e $\lfloor x \rfloor \leq x$, seguem diretamente. Para as outras duas desigualdades, escreva $x = \lceil x \rceil - \epsilon'$, em que $0 \leq \epsilon' < 1$. Então, $0 \leq \lceil x \rceil - x < 1$, e segue a desigualdade desejada. **47. a)** Se $x < n$, como $\lfloor x \rfloor \leq x$, segue que $\lfloor x \rfloor < n$. Suponha que $x \geq n$. Pela definição de função chão, segue que $\lfloor x \rfloor \geq n$. Isto significa que, se $\lfloor x \rfloor < n$, então $x < n$. **b)** Se $n < x$, então como $x \leq \lceil x \rceil$, segue que $n \leq \lceil x \rceil$. Suponha que $n \geq x$. Pela definição de função teto, segue que $\lceil x \rceil \leq n$. Isto significa que, se $n < \lceil x \rceil$, então $n < x$. **49.** Se n for par, então $n = 2k$ para algum inteiro k. Assim, $\lfloor n/2 \rfloor = \lfloor k \rfloor = k = n/2$. Se n for ímpar, então $n = 2k + 1$ para algum inteiro k. Assim, $\lfloor n/2 \rfloor = \lfloor k + \frac{1}{2} \rfloor = k = (n-1)/2$. **51.** Suponha que $x \geq 0$. O lado esquerdo é $\lceil -x \rceil$ e o lado direito é $-\lfloor x \rfloor$. Se x for um inteiro, então ambos os lados são iguais a $-x$. Caso contrário, seja $x = n + \epsilon$, em que n é um número natural e ϵ é um número real com $0 \leq \epsilon < 1$. Então, $\lceil -x \rceil = \lceil -n - \epsilon \rceil = -n$ e $-\lfloor x \rfloor = -\lfloor n + \epsilon \rfloor = -n$ também. Quando $x < 0$, a equação também é válida, pois ela pode ser obtida substituindo $-x$ por x. **53.** $\lceil b \rceil - \lfloor a \rfloor - 1$ **55. a)** 1 **b)** 3 **c)** 126 **d)** 3 600 **57. a)** 100 **b)** 256 **c)** 1 030 **d)** 30 200

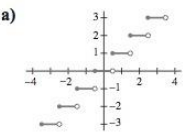
59.



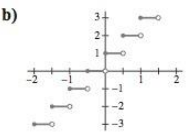
61.



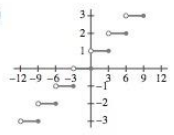
63. a)



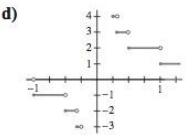
b)



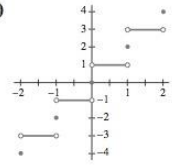
c)



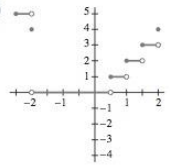
d)



e)



f)



g) Veja a parte (a). **65.** $f^{-1}(y) = (y - 1)^{1/3}$ **67. a)** $f_{A\cap B}(x) = 1 \leftrightarrow x \in A \cap B \leftrightarrow x \in A$ e $x \in B \leftrightarrow f_A(x) = 1$ e $f_B(x) = 1 \leftrightarrow f_A(x) f_B(x) = 1$ **b)** $f_{A\cup B}(x) = 1 \leftrightarrow x \in A \cup B \leftrightarrow x \in A$ ou $x \in B \leftrightarrow f_A(x) = 1$ ou $f_B(x) = 1 \leftrightarrow f_A(x) + f_B(x) - f_A(x) f_B(x) = 1$ **c)** $f_{\overline{A}}(x) = 1 \leftrightarrow x \in \overline{A} \leftrightarrow x \notin A \leftrightarrow f_A(x) = 0 \leftrightarrow 1 - f_A(x) = 1$ **d)** $f_{A\oplus B}(x) = 1 \leftrightarrow x \in A \oplus B \leftrightarrow (x \in A$ e $x \notin B)$ ou $(x \notin A$ e $x \in B) \leftrightarrow f_A(x) + f_B(x) - 2 f_A(x) f_B(x) = 1$. **69. a)** Verdadeira; como $\lfloor x \rfloor$ já é um inteiro, $\lceil \lfloor x \rfloor \rceil = \lfloor x \rfloor$. **b)** Falsa; $x = \frac{1}{2}$ é um contra-exemplo. **c)** Verdadeira; se x ou y for um inteiro, então, pela propriedade 4b na Tabela 1, a diferença é 0. Se nem x nem y for um inteiro, então $x = n + \epsilon$ e $y = m + \delta$, em que n e m são inteiros e ϵ e δ são números reais positivos menores do que 1. Então, $m + n < x + y < m + n + 2$, de modo que $\lceil x + y \rceil$ ou é $m + n + 1$ ou $m + n + 2$. Logo, a expressão dada ou é $(n + 1) + (m + 1) - (m + n + 1) = 1$ ou $(n + 1) + (m + 1) - (m + n + 2) = 0$, como desejado. **d)** Falsa; $x = \frac{1}{4}$ e $y = 3$ é um contra-exemplo. **e)** Falsa; $x = \frac{1}{2}$ é um contra-exemplo. **71. a)** Se x for um inteiro positivo, então os dois lados são iguais. Assim, suponha que $x = n^2 + m + \epsilon$, em que n^2 é o maior quadrado perfeito menor do que x, m é um inteiro não negativo e $0 < \epsilon \leq 1$. Então, ambos, $\sqrt{x}$ e $\sqrt{\lfloor x \rfloor} = \sqrt{n^2 + m}$, estão entre n e $n + 1$, de modo que ambos os lados são iguais a n. **b)** Se x for um inteiro positivo, então os dois lados são iguais. Assim, suponha que $x = n^2 - m - \epsilon$, em que n^2 é o menor quadrado perfeito maior do que x, m é um inteiro não negativo e ϵ é um número real com $0 < \epsilon \leq 1$. Então, ambos, $\sqrt{x}$ e $\sqrt{\lfloor x \rfloor} = \sqrt{n^2 - m}$, estão entre $n - 1$ e n. Portanto, ambos os lados da equação são iguais a n. **73. a)** Domínio é $\mathbf{Z}$; contradomínio é $\mathbf{R}$; domínio de definição é o conjunto dos inteiros não nulos; conjunto de valores para os quais f não está definida é $\{0\}$; não é uma função completa. **b)** Domínio é $\mathbf{Z}$; contradomínio é $\mathbf{Z}$; domínio de definição é $\mathbf{Z}$; conjunto de valores para os quais f não está definida é $\emptyset$; função completa. **c)** Domínio é $\mathbf{Z} \times \mathbf{Z}$; contradomínio é $\mathbf{Q}$; domínio de definição é $\mathbf{Z} \times (\mathbf{Z} - \{0\})$; conjunto de valores para os quais f não está definida é $\mathbf{Z} \times \{0\}$; não é uma função completa. **d)** Domínio é $\mathbf{Z} \times \mathbf{Z}$; contradomínio é $\mathbf{Z}$; domínio de definição é $\mathbf{Z} \times \mathbf{Z}$; conjunto de valores para os quais f não está definida é $\emptyset$; função completa. **e)** Domínio é $\mathbf{Z} \times \mathbf{Z}$; contradomínio é $\mathbf{Z}$; domínio de definição é $\{(m, n) \mid m > n\}$; conjunto de valores para os quais f não está definida é $\{(m, n) \mid m \leq n\}$; não é uma função completa. **75. a)** Por definição, dizer que S tem cardinalidade m é dizer que S tem exatamente m elementos distintos. Portanto, podemos associar o primeiro elemento a 1, o segundo a 2, e assim por diante. Isto fornece uma bijeção. **b)** Pela parte (a), existe uma bijeção f de S para $\{1, 2, \ldots, m\}$ e uma bijeção g de T para $\{1, 2, \ldots, m\}$. Então, a composição $g^{-1} \circ f$ é a bijeção desejada de S para T. **77.** É claro a partir da fórmula que o intervalo de valores que a função assume para um valor fixo de $m + n$, digamos $m + n = x$, é $(x - 2)(x - 1)/2 + 1$ a $(x - 2)(x - 1)/2 + (x - 1)$, pois m pode assumir os valores 1, 2, 3, …, $(x - 1)$ sob estas condições, e o primeiro termo na fórmula é um inteiro positivo fixo quando $m + n$ está fixo. Para mostrar que esta função é injetora e sobrejetiva, precisamos simplesmente mostrar que o intervalo de valores para $x + 1$ começa precisamente onde o intervalo de valores para x terminou, isto é, que $f(x - 1, 1) + 1 = f(1, x)$. Temos $f(x - 1, 1) + 1 = \frac{(x-2)(x-1)}{2} + (x - 1) + 1 = \frac{x^2-x+2}{2} = \frac{(x-1)x}{2} + 1 = f(1, x)$.

Seção 2. 4

1. a) 3 **b)** −1 **c)** 787 **d)** 2639 **3. a)** $a_0 = 2, a_1 = 3, a_2 = 5, a_3 = 9$ **b)** $a_0 = 1, a_1 = 4, a_2 = 27, a_3 = 256$ **c)** $a_0 = 0, a_1 = 0, a_2 = 1, a_3 = 1$ **d)** $a_0 = 0, a_1 = 1, a_2 = 2, a_3 = 3$ **5. a)** 2, 5, 8, 11, 14, 17, 20, 23, 26, 29 **b)** 1, 1, 1, 2, 2, 2, 3, 3, 3, 4 **c)** 1, 1, 3, 3, 5, 5, 7, 7, 9, 9 **d)** −1, −2, −2, 8, 88, 656, 4912, 40064, 362368, 3627776 **e)** 3, 6, 12, 24, 48, 96, 192, 384, 768, 1536 **f)** 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 **g)** 1, 2, 2, 3, 3, 3, 3, 4, 4, 4 **h)** 3, 3, 5, 4, 4, 3, 5, 5, 4, 3 **7.** Cada termo pode ser duas vezes o termo anterior; o n-ésimo termo poderia ser obtido a partir do termo anterior somando $n - 1$; os termos poderiam ser inteiros positivos que não são múltiplos de 3; existem infinitas outras possibilidades. **9. a)** Um 1 e um 0, seguidos por dois 1s e dois 0s, seguidos por três 1s e três 0s, e assim por diante; 1, 1, 1 **b)** Os inteiros positivos estão listados em ordem crescente com cada inteiro positivo par listado duas vezes; 9, 10, 10. **c)** Os termos em posições com numeração ímpar são as potências sucessivas de 2; os termos nas posições com numeração par são todos 0; 32, 0, 64. **d)** $a_n = 3 \cdot 2^{n-1}$; 384, 768, 1536 **e)** $a_n = 15 - 7(n - 1) = 22 - 7n$; −34, −41, −48 **f)** $a_n = (n^2 + n + 4)/2$; 57, 68, 80 **g)** $a_n = 2n^3$; 1024, 1458, 2000 **h)** $a_n = n! + 1$; 362881, 3628801, 39916801 **11.** Entre os inteiros 1, 2, …, a_n, em que a_n é o n-ésimo inteiro positivo que não é um quadrado perfeito, os não quadrados são $a_1, a_2, \ldots, a_n$ e os quadrados são $1^2, 2^2, \ldots, k^2$, em que k é o inteiro com $k^2 < n + k < (k + 1)^2$. Conseqüentemente, $a_n = n + k$, em que $k^2 < a_n < (k + 1)^2$. Para encontrar k, observe primeiro que $k^2 < n + k < (k + 1)^2$, de modo que $k^2 + 1 \leq n + k \leq (k + 1)^2 - 1$. Portanto, $(k - \frac{1}{2})^2 + \frac{3}{4} = k^2 - k + 1 \leq n \leq k^2 + k = (k + \frac{1}{2})^2 - \frac{1}{4}$. Segue que $k - \frac{1}{2} < \sqrt{n} < k + \frac{1}{2}$, de modo que $k = \{\sqrt{n}\}$ e $a_n = n + k = n + \{\sqrt{n}\}$. **13. a)** 20 **b)** 11 **c)** 30 **d)** 511 **15. a)** 1533 **b)** 510 **c)** 4923 **d)** 9842 **17. a)** 21 **b)** 78 **c)** 18 **d)** 18 **19.** $\sum_{j=1}^{n} (a_j - a_{j-1}) = a_n - a_0$ **21. a)** n^2 **b)** $n(n + 1)/2$ **23.** 15150 **25.** $\frac{n(n+1)(2n+1)}{3} + \frac{n(n+1)}{2} + (n + 1)(m - (n + 1)^2 + 1)$, em que $n = \lfloor \sqrt{m} \rfloor - 1$ **27. a)** 0 **b)** 1680 **c)** 1 **d)** 1024 **29.** 34 **31. a)** Contável, −1, −2, −3, −4,… **b)** Contável, 0, 2, −2, 4, −4, … **c)** Incontável **d)** Contável, 0, 7, −7, 14, −14, … **33. a)** Contável: associe n com a seqüência de

x)) **h)** $\exists x \exists y (x \neq y \wedge L(\text{Lynn}, x) \wedge L(\text{Lynn}, y) \wedge \forall z (L(\text{Lynn}, z) \rightarrow (z = x \vee z = y)))$ **i)** $\forall x\, L(x, x)$ **j)** $\exists x\, \forall y (L(x, y) \leftrightarrow x = y)$ **11. a)** A(Lois, professor Michaels) **b)** $\forall x(S(x) \rightarrow A(x, \text{professor Gross}))$ **c)** $\forall x(F(x) \rightarrow (A(x, \text{professor Miller}) \vee A(\text{professor Miller}, x)))$ **d)** $\exists x(S(x) \wedge \forall y(F(y) \rightarrow \neg A(x, y)))$ **e)** $\exists x(F(x) \wedge \forall y(S(y) \rightarrow \neg A(y, x)))$ **f)** $\forall y(F(y) \rightarrow \exists x(S(x) \vee A(x, y)))$ **g)** $\exists x(F(x) \wedge \forall y((F(y) \wedge (y \neq x)) A(x, y)))$ **h)** $\exists x(S(x) \wedge \forall y(F(y) \rightarrow \neg A(y, x)))$ **13. a)** $\neg M$(Chou, Koko) **b)** $\neg M(\text{Arlene}, \text{Sarah}) \wedge \neg T(\text{Arlene}, \text{Sarah})$ **c)** $\neg M$(Deborah, Jose) **d)** $\forall x M(x,$ Ken) **e)** $\forall x \neg T$ (x, Nina) **f)** $\forall x(T\text{-}x, \text{Avi}) \vee M(x, \text{Avi}))$ **g)** $\exists x \forall y (y \neq x \rightarrow M(x, y))$ **h)** $\exists x \forall y (y \neq x \rightarrow (M(x, y) \vee T(x, y)))$ **i)** $\exists x \exists y (x \neq y \wedge M(x, y) \wedge M(y, x))$ **j)** $\exists x M(x, x)$ **k)** $\exists x \forall y\, (x \neq y \rightarrow (\neg M(x, y) \wedge \neg T(y, x)))$ **l)** $\forall x (\exists y (x \neq y \wedge (M(y, x) \vee T(y, x))))$ **m)** $\exists x \exists y (x \neq y \wedge M(x, y) \wedge T(y, x))$ **n)** $\exists x \exists y (x \neq y \wedge \forall z((z \neq x \wedge z \neq y) \rightarrow (M(x, z) \vee M(y, z) \vee T(x, z) \vee T(y, z))))$ **15. a)** $\forall x\, P(x)$, em que $P(x)$ é "x precisa de um curso de matemática discreta" e o domínio consiste em todos os estudantes de ciência da computação. **b)** $\exists x\, P(x)$, em que $P(x)$ é "x possui seu próprio computador" e o domínio consiste em todos os estudantes nesta sala. **c)** $\forall x \exists y P(x, y)$, em que $P(x, y)$ é "x participou do curso y", o domínio para x consiste em todos os estudantes nesta sala, e o domínio para y consiste em todos os cursos de ciência da computação. **d)** $\exists x \exists y P(x, y)$, em que $P(x, y)$ e domínios são os mesmos da parte (c). **e)** $\forall x \forall y P(x, y)$, em que $P(x, y)$ é "x já esteve em y", o domínio para x consiste em todos os estudantes desta sala, e o domínio para y consiste em todos os prédios do campus. **f)** $\exists x \exists y \forall z (P(z, y) \rightarrow Q(x, z))$, em que $P(z, y)$ é "z está em y" e $Q(x, z)$ é "x já esteve em z"; o domínio para x consiste em todos os estudantes na sala, o domínio para y consiste em todos os prédios do campus, e o domínio para z consiste em todas as salas. **g)** $\forall x \forall y \exists z (P(z, y) \wedge Q(x, z))$, com o mesmo ambiente que na parte (f) **17. a)** $\forall u \exists m (A(u, m) \wedge \forall n (n \neq m \rightarrow \neg A(u, n)))$, em que $A(u, m)$ significa que o usuário u tem acesso à caixa de entrada m **b)** $\exists p \forall e (H(e) \wedge S(p,$ executando$)) \rightarrow S$ (kernel, funcionando corretamente), em que $H(e)$ significa que a condição de erro e está em vigor e $S(x, y)$ significa que o *status* de x é y **c)** $\forall u \forall s (E(s,$ .edu$) \rightarrow A(u, s))$, em que $E(s, x)$ significa que o site da Web s tem a extensão x, e $A(u, s)$ significa que o usuário u pode acessar o site da Web s **d)** $\exists x \exists y (x \neq y \wedge \forall z ((\forall s\, M(z, s)) \leftrightarrow (z = x \vee z = y)))$, em que $M(a, b)$ significa que o sistema a monitora o servidor remoto b **19. a)** $\forall x \forall y ((x < 0) \wedge (y < 0) \rightarrow (x + y < 0))$ **b)** $\neg \forall x \forall y ((x > 0) \wedge (y > 0) \rightarrow (x - y > 0))$ **c)** $\forall x \forall y\, (x^2 + y^2 \geq (x + y)^2)$ **d)** $\forall x \forall y\ (|xy| = |x||y|)$ **21.** $\forall x \exists a \exists b \exists c \exists d\, ((x > 0) \rightarrow x = a^2 + b^2 + c^2 + d^2)$, em que o domínio consiste em todos os inteiros **23. a)** $\forall x \forall y\, ((x < 0) \wedge (y < 0) \rightarrow (xy > 0))$ **b)** $\forall x (x - x = 0)$ **c)** $\forall x \exists a \exists b (a \neq b \wedge \forall c (c^2 = x \leftrightarrow (c = a \vee c = b)))$ **d)** $\forall x((x < 0) \rightarrow \neg \exists y (x = y^2))$ **25. a)** Existe um elemento neutro da multiplicação para os reais. **b)** O produto de dois números reais negativos é sempre um número real positivo. **c)** Existem números reais x e y tais que x^2 excede y, mas x é menor do que y. **d)** Os números reais são fechados pela operação de adição. **27. a)** Verdadeira **b)** Verdadeira **c)** Verdadeira **d)** Verdadeira **e)** Verdadeira **f)** Falsa **g)** Falsa **h)** Verdadeira **i)** Falsa **29. a)** $P(1, 1) \wedge P(1, 2) \wedge P(1, 3) \wedge P(2, 1) \wedge P(2, 2) \wedge P(2, 3) \wedge P(3, 1) \wedge P(3, 2) \wedge P(3, 3)$ **b)** $P(1, 1) \vee P(1, 2) \vee P(1, 3) \vee P(2, 1) \vee P(2, 2) \vee P(2, 3) \vee P(3, 1) \vee P(3, 2) \vee P(3, 3)$ **c)** $(P(1, 1) \wedge P(1, 2) \wedge P(1, 3)) \vee (P(2, 1) \wedge P(2, 2) \wedge P(2, 3)) \vee (P(3, 1) \wedge P(3, 2) \wedge P(3, 3))$ **d)** $(P(1, 1) \vee P(2, 1) \vee P(3, 1)) \wedge (P(1, 2) \vee P(2, 2) \vee P(3, 2)) \wedge (P(1, 3) \vee P(2, 3) \vee P(3, 3))$ **31. a)** $\exists x \forall y \exists z \neg T(x, y, z)$ **b)** $\exists x \forall y \neg P(x, y) \wedge \exists x \forall y \neg Q(x, y)$ **c)** $\exists x \forall y\, (\neg P(x, y) \vee \forall z \neg R(x, y, z))$ **d)** $\exists x \forall y (P(x, y) \wedge \neg Q(x, y))$ **33. a)** $\exists x \exists y \neg P(x, y)$ **b)** $\exists y \forall x \neg P(x, y)$ **c)** $\exists y \exists x (\neg P(x, y) \wedge \neg Q(x, y))$ **d)** $(\forall x \forall y P(x, y)) \vee (\exists x \exists y \neg Q(x, y))$ **e)** $\exists x (\forall y \exists z \neg P(x, y, z) \vee \forall z \exists y \neg P(x, y, z))$ **35.** Qualquer domínio com quatro ou mais membros torna a afirmação verdadeira; qualquer domínio com três ou menos membros torna a afirmação falsa. **37. a)** Existe alguém nesta sala tal que para quaisquer dois cursos de matemática diferentes, estes não são os dois únicos cursos de matemática que a pessoa freqüentou. **b)** Qualquer pessoa ou visitou a Líbia ou não visitou um pais diferente da Líbia. **c)** Alguém escalou todas as montanhas do Himalaia. **d)** Existe alguém que nunca participou de um filme ao lado de Kevin Bacon nem contracenou com alguém que já filmou com Kevin Bacon. **39. a)** $x = 2$, $y = -2$ **b)** $x = -4$ **c)** $x = 17, y = -1$ **41.** $\forall x \forall y \forall z ((x \cdot y) \cdot z = x \cdot (y \cdot z))$ **43.** $\forall m \forall b (m \neq 0 \rightarrow \exists x (mx + b = 0 \wedge \forall w (mw + b = 0 \rightarrow w = x)))$ **45. a)** Verdadeira **b)** Falsa **c)** Verdadeira **47.** $\neg(\exists x \forall y P(x, y)) \leftrightarrow \forall x(\neg \forall y P(x, y)) \leftrightarrow \forall x \exists y \neg P(x, y)$ **49. a)** Suponha que $\forall x\, P(x) \wedge \exists x Q(x)$ seja verdadeira. Então, $P(x)$ é verdadeira para todo x e existe um elemento y para o qual $Q(y)$ é verdadeira. Como $P(x) \wedge Q(y)$ é verdadeira para todo x e existe um y para o qual $Q(y)$ é verdadeira, $\forall x \exists y (P(x) \wedge Q(y))$ é verdadeira. Reciprocamente, suponha que a segunda proposição seja verdadeira. Seja x um elemento no domínio. Existe um y tal que $Q(y)$ é verdadeira, de modo que $\exists x Q(x)$ é verdadeira. Como $\forall x\, P(x)$ também é verdadeira, segue que a primeira proposição é verdadeira. **b)** Suponha que $\forall x\, P(x) \vee \exists x Q(x)$ seja verdadeira. Então, ou $P(x)$ é verdadeira para todo x, ou existe um y para o qual $Q(y)$ é verdadeira. No primeiro caso, $P(x) \vee Q(y)$ é verdadeira para todo x, de modo que $\forall x \exists y (P(x) \vee Q(y))$ é verdadeira. No segundo caso, $Q(y)$ é verdadeira para um y particular, de modo que $P(x) \vee Q(y)$ é verdadeira para todo x e, conseqüentemente, $\forall x \exists y (P(x) \vee Q(y))$ é verdadeira. Reciprocamente, suponha que a segunda proposição seja verdadeira. Se $P(x)$ for verdadeira para todo x, então a primeira proposição é verdadeira. Caso contrário, $P(x)$ é falsa para algum x, e para este x deve existir um y tal que $P(x) \vee Q(y)$ é verdadeira. Portanto, $Q(y)$ deve ser verdadeira, de modo que $\exists y Q(y)$ é verdadeira. Segue que a primeira proposição deve ser válida. **51.** Mostraremos como uma expressão pode ser posta na forma normal prenex (FNP) se subexpressões nela puderem ser colocadas na FNP. Então, trabalhando de dentro para fora, qualquer expressão pode ser posta em FNP. (Para formalizar o argumento, é necessário usar o método de indução estrutural que será discutido na Seção 4.3.) Pelo Exercício 45 da Seção 1.2, podemos supor que a proposição usa apenas $\vee$ e $\neg$ como conectores lógicos. Observe agora que qualquer expressão sem quantificadores já está em FNP. (Este é o caso base da argumentação.) Suponha agora que a proposição seja da forma $Qx\, P(x)$, em que Q é um quantificador. Como $P(x)$ é uma expressão mais curta que a proposição original, podemos colocá-la em FNP. Então, Qx seguida por esta PNF está novamente em FNP e é equivalente à proposição original. A seguir, suponha que a proposição seja da forma $\neg P$. Se P já estiver em FNP, passamos o sinal de negação por todos os quantificadores usando as equivalências da Tabela 2 da Seção 1.3. Finalmente, suponha que a proposição seja da forma $P \vee$

9. **procedure** *verifica palindromo* ($a_1a_2 \ldots a_n$: seqüência)
resposta := **true**
for $i := 1$ **to** $\lfloor n/2 \rfloor$
if $a_i \neq a_{n+1-i}$ **then** *resposta* := **false**
end {a *resposta* é verdadeira se e somente se a seqüência for um palíndromo}

11. **procedure** *troca*(x, y: números reais)
$z := x$
$x := y$
$y := z$

O número mínimo de proposições necessárias é três. **13.** Busca linear: $i := 1$, $i := 2$, $i := 3$, $i := 4$, $i := 5$, $i := 6$, $i := 7$, *posicao* := 7; busca binária: $i := 1, j := 8, m := 4$, $i := 5, m := 6, i := 7, m := 7, j := 7$, *posicao* := 7

15. **procedure** *insira*(x, a_1, a_2, …, a_n: inteiros)
{a lista está em ordem: $a_1 \leq a_2 \leq \cdots \leq a_n$}
$a_{n+1} := x + 1$
$i := 1$
while $x > a_i$
$i := i + 1$
for $j := 0$ **to** $n - i$
$a_{n-j+1} := a_{n-j}$
$a_i := x$
{x foi inserido na posição correta}

17. **procedure** *primeiro maior*(a_1, …, a_n: inteiros)
max := a_1
posicao := 1
for $i := 2$ **to** n
begin
if *max* < a_i **then**
begin
max := a_i
posicao := i
end
end

19. **procedure** *media-mediano-max-min*(a, b, c: inteiros)
media := $(a + b + c)/3$
{as seis ordens diferentes de a, b, c com relação a $\geq$ serão tratadas separadamente}
if $a > b$ **then**
begin
if $b > c$ **then**
mediano := b; *max* := a; *min* := c
end
…
(O restante do algoritmo é análogo.)

21. **procedure** *primeiros-tres* (a_1, a_2, …, a_n: inteiros)
if $a_1 > a_2$ **then** troque a_1 e a_2
if $a_2 > a_3$ **then** troque a_2 e a_3
if $a_1 > a_2$ **then** troque a_1 e a_2

23. **procedure** *sobrejetiva*(f: função de A em B, em que $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_m\}$, $a_1, \ldots, a_n$, $b_1, \ldots, b_m$ são inteiros)
for $i := 1$ **to** m
atingiu(b_i) := 0
contador := 0
for $j := 1$ **to** n
if *atingiu*($f(a_j)$) = 0 **then**
begin
atingiu($f(a_j)$) := 1
contador := *contador* + 1
end
if *contador* = m **then** *sobrejetiva* := **true**
else *sobrejetiva* := **false**

25. **procedure** *uns*(a: cadeia de bits, $a = a_1a_2 \ldots a_n$)
uns:= 0
for $i := 1$ **to** n
begin
if $a_i := 1$ **then**
uns := *uns* + 1
end {*uns* é o número de uns na cadeia de bits a}

27. **procedure** *busca ternaria*(s: inteiro, a_1, a_2, …, a_n: inteiros crescentes)
$i := 1$
$j := n$
while $i < j - 1$
begin
$l = \lfloor (i + j)/3 \rfloor$
$u = \lfloor 2(i + j)/3 \rfloor$
if $x > a_u$ **then** $i := u + 1$
else if $x > a_l$ **then**
begin
$i := l + 1$
$j := u$
end
else $j := l$
end
if $x = a_i$ **then** *posicao* := i
else if $x = a_j$ **then** *posicao* := j
else *posicao* := 0
{*posicao* é o subscrito do termo igual a x (0 se não for encontrado)}

29. **procedure** *encontre uma moda*(a_1, a_2, …, a_n: inteiros não decrescentes)
contadordamoda := 0
$i := 1$
while $i \leq n$
begin
valor := a_i
contador := 1
while $i \leq n$ e a_i = *valor*
begin
contador := *contador* + 1
$i := i + 1$
end
if *contador* > *contadordamoda* **then**
begin
contadordamoda := *contador*
moda := *valor*
end
end
{*moda* é o primeiro valor que ocorre mais freqüentemente}

31. **procedure** *encontre duplicados*(a_1, a_2, ..., a_n: inteiros)
posicao := 0
$i := 2$
while $i \leq n$ e *posicao* $= 0$
begin
$j := 1$
while $j < i$ e *posicao* $= 0$
if $a_i = a_j$ **then** *posicao* := i
else $j := j + 1$
$i := i + 1$
end
{*posicao* é o subscrito do primeiro valor que repete um valor prévio na seqüência}

33. **procedure** *encontre decrescimo* (a_1, a_2, ..., a_n: inteiros positivos)
posicao := 0
$i := 2$
while $i \leq n$ e *posicao* $= 0$
if $a_i < a_{i-1}$ **then** *posicao* := i
else $i := i + 1$
{*posicao* é o subscrito do primeiro valor menor do que o imediatamente precedente}

35. No final da primeira passagem: 1, 3, 5, 4, 7; no final da segunda passagem: 1, 3, 4, 5, 7; no final da terceira passagem: 1, 3, 4, 5, 7; no final da quarta passagem: 1, 3, 4, 5, 7

37. **procedure** *bubblesort melhor* (a_1, ..., a_n: inteiros)
$i := 1$; *acabado* := **false**
while ($i < n$ e *acabado* = **false**)
begin
acabado := **true**
for $j := 1$ **to** $n - i$
if $a_j > a_{j+1}$ **then**
begin
troque a_j e a_{j+1}
acabado := **false**
end
$i := i + 1$
end {a_1, ..., a_n está em ordem crescente}

39. No final das primeira, segunda e terceira passagens: 1, 3, 5, 7, 4; no final da quarta passagem: 1, 3, 4, 5, 7 **41.** **a)** 1, 5, 4, 3, 2; 1, 2, 4, 3, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5 **b)** 1, 4, 3, 2, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5 **c)** 1, 2, 3, 4, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5 **43.** Nós executamos o algoritmo de busca linear dado como Algoritmo 2 nesta seção, exceto que substituímos $x \neq a_i$ por $x < a_i$, e substituímos a cláusula **else** por **else** *posicao* := $n + 1$. **45.** $2 + 3 + 4 + \cdots + n = (n^2 + n - 2)/2$ **47.** Encontre a posição do 2 na lista 3 (uma comparação) e insira-o na frente do 3, de modo que a lista agora é 2, 3, 4, 5, 1, 6. Encontre a posição do 4 (compare-o com o 2 e depois com o 3) e insira-o, deixando 2, 3, 4, 5, 1, 6. Encontre a posição para o 5 (compare-o com o 3 e então com o 4) e insira-o, deixando 2, 3, 4, 5, 1, 6. Encontre a posição para o 1 (compare-o com o 3, depois com o 2 e então com o 2 novamente) e insira-o, deixando 1, 2, 3, 4, 5, 6. Encontre a posição para o 6 (compare-o com o 3, depois com o 4 e então com o 5) e insira-o, dando a resposta final 1, 2, 3, 4, 5, 6.

49. **procedure** *binary insertion sort*(a_1, a_2, ..., a_n: números reais com $n \geq 2$)
for $j := 2$ **to** n
begin
{busca binária para inserir a posição i}
esquerda := 1
direita := $j - 1$
while *esquerda* < *direita*
begin
medio := $\lfloor (esquerda + direita)/2 \rfloor$
if $a_j > a_{medio}$ **then** *esquerda* := *medio* + 1
else *direita* := *medio*
end
if $a_j < a_{esquerda}$ **then** $i :=$ *esquerda* **else** $i :=$ *esquerda* + 1
{insira a_j na posição i movendo a_i até a_{j-1} em direção ao final da lista}
$m := a_j$
for $k := 0$ **to** $j - i - 1$
$a_{j-k} := a_{j-k-1}$
$a_i := m$
end {$a_1, a_2, \ldots, a_n$ está ordenado}

51. A variação do Exercício 50. **53.** **a)** Duas de vinte e cinco centavos, uma de um centavo. **b)** Duas de vinte e cinco centavos, uma de dez, uma de cinco, quatro de um. **c)** Três de vinte e cinco centavos, uma de um. **d)** Duas de vinte e cinco centavos, uma de dez. **55.** O algoritmo voraz usa menos moedas nas partes (a), (c), e (d). **a)** Duas de vinte e cinco centavos, uma de um. **b)** Duas de vinte e cinco centavos, uma de dez, nove de um. **c)** Três de vinte e cinco centavos, uma de um. **d)** Duas de vinte e cinco centavos, uma de dez. **57.** **a)**A variável f dará o instante de encerramento da última palestra selecionada, começando com f igual ao instante em que o auditório ficou disponível. Ordene as palestras em ordem crescente de horários de encerramento e comece no topo da lista. Em cada estágio do algoritmo, vá para baixo na lista de palestras a partir de onde você parou e encontre a primeira cujo horário de início não é menor do que f. Marque esta palestra e atualize f para registrar seu horário de encerramento. **b)** A palestra de 9:00–9:45, a palestra de 9:50–10:15, a palestra de 10:15–10:45, a palestra de 11:00–11:15. **59.** **a)** Aqui, suporemos que os homens sejam os pretendentes e as mulheres as pretendidas.
procedure *estavel*(M_1, M_2, ..., M_s, W_1, W_2, ..., W_s:
listas de preferência)
for $i := 1$ **to** s
marque o homem i como rejeitado
for $i := 1$ **to** s
tome a lista de rejeição do homem i como vazia
for $j := 1$ **to** s
tome a lista de propostas da mulher j como vazia
while ainda restarem homens rejeitados
begin
for $i := 1$ **to** s
if o homem i estiver marcado como rejeitado **then** adicione i à lista de propostas da mulher j que estiver em posição mais alta em sua lista de preferência, mas não apareça em sua lista de rejeição, e marque i como não rejeitado

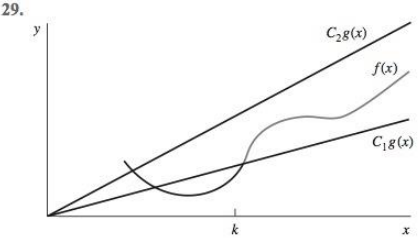
for $j := 1$ **to** s
  if a lista de propostas da mulher j for não vazia **then**
  remova da lista de propostas de j todos os homens i, exceto o homem i_0 que estiver na posição mais alta em sua lista de preferência, e para cada um desses homens i, marque-o como rejeitado e adicione j à sua lista de rejeição
end
for $j := 1$ **to** s
  associe j ao único homem na lista de propostas de j
{Esta associação é estável.}

b) Existem no máximo s^2 iterações do laço **while**, de modo que o algoritmo deve terminar. De fato, se na conclusão do laço **while** restar algum homem rejeitado, então algum homem deve ter sido rejeitado e assim sua lista de rejeição cresceu. Assim, em cada passagem pelo laço **while**, pelo menos mais uma das s^2 rejeições possíveis terá sido registrada, a menos que o laço esteja para acabar. Além disso, quando o laço **while** termina, cada homem terá uma proposta pendente e cada mulher terá no máximo uma proposta pendente, de modo que a associação deve ser injetora. **c)** Se a associação não for estável, então existem um homem m e uma mulher w tal que m prefere w à mulher w' com quem está associado, e w prefere m ao homem com quem está associada. Mas m deve ter feito uma proposta a w antes de ter feito a w', pois ele prefere a primeira. Como m não acabou associado a w, ela deve tê-lo rejeitado. As mulheres rejeitam um pretendente apenas quando têm uma proposta melhor, e elas eventualmente ficam associadas a um pretendente pendente, de modo que o homem com quem w está associada deve ser melhor a seus olhos do que m, contradizendo nossa hipótese original. Portanto, o casamento é estável. **61.** Execute os dois programas com suas entradas simultaneamente e registre qual deles pára.

Seção 3.2

1. As escolhas de C e k não são únicas. **a)** $C = 1$, $k = 10$ **b)** $C = 4$, $k = 7$ **c)** Não. **d)** $C = 5$, $k = 1$ **e)** $C = 1$, $k = 0$ **f)** $C = 1$, $k = 2$ **3.** $x^4 + 9x^3 + 4x + 7 \leq 4x^4$ para todo $x > 9$; parâmetros $C = 4$, $k = 9$ **5.** $(x^2 + 1)/(x + 1) = x - 1 + 2/(x + 1) < x$ para todo $x > 1$; parâmetros $C = 1$, $k = 1$ **7.** As escolhas de C e k não são únicas. **a)** $n = 3$, $C = 3$, $k = 1$ **b)** $n = 3$, $C = 4$, $k = 1$ **c)** $n = 1$, $C = 2$, $k = 1$ **d)** $n = 0$, $C = 2$, $k = 1$ **9.** $x^2 + 4x + 17 \leq 3x^3$ para todo $x > 17$, de modo que $x^2 + 4x + 17$ é $O(x^3)$, com parâmetros $C = 3$, $k = 17$. Entretanto, se x^3 fosse $O(x^2 + 4x + 17)$, então $x^3 \leq C(x^2 + 4x + 17) \leq 3Cx^2$ para algum C, para todo x suficientemente grande, o que implica que $x \leq 3C$ para todo x suficientemente grande, o que é impossível. Portanto, x^3 não é $O(x^2 + 4x + 17)$. **11.** $3x^4 + 1 \leq 4x^4 = 8(x^4/2)$ para todo $x > 1$, de modo que $3x^4 + 1$ é $O(x^4/2)$, com parâmetros $C = 8$, $k = 1$. Além disso, $x^4/2 \leq 3x^4 + 1$ para todo $x > 0$, de modo que $x^4/2$ é $O(3x^4 + 1)$, com parâmetros $C = 1$, $k = 0$. **13.** Como $2^n \leq 3^n$ para todo $n > 0$, segue que 2^n é $O(3^n)$, com parâmetros $C = 1$, $k = 0$. Entretanto, se 3^n fosse $O(2^n)$, então para algum C, $3^n \leq C \cdot 2^n$ para todo n suficientemente grande. Isto diz que $C \geq (3/2)^n$ para todo n suficientemente grande, o que é impossível. Logo, 3^n não é $O(2^n)$. **15.** Todas as funções para as quais existem números reais k e C com $|f(x)| \leq C$ para $x > k$. Estas são as funções $f(x)$ que são limitadas para todo x suficientemente grande. **17.** Existem constantes C_1, C_2, k_1 e k_2 tais que $|f(x)| \leq C_1|g(x)|$ para todo $x > k_1$ e $|g(x)| \leq C_2|h(x)|$ para todo $x > k_2$. Logo, para $x > \max(k_1, k_2)$ segue que $|f(x)| \leq C_1|g(x)| \leq C_1C_2|h(x)|$. Isto mostra que $f(x)$ é $O(h(x))$. **19. a)** $O(n^3)$ **b)** $O(n^5)$ **c)** $O(n^3 \cdot n!)$ **21. a)** $O(n^2 \log n)$ **b)** $O(n^2(\log n)^2)$ **c)** $O(n^{2^n})$ **23. a)** Nem $\Theta(x^2)$ nem $\Omega(x^2)$ **b)** $\Theta(x^2)$ e $\Omega(x^2)$ **c)** Nem $\Theta(x^2)$ nem $\Omega(x^2)$ **d)** $\Omega(x^2)$, mas não $\Theta(x^2)$ **e)** $\Omega(x^2)$, mas não $\Theta(x^2)$ **f)** $\Omega(x^2)$ e $\Theta(x^2)$ **25.** Se $f(x)$ for $\Theta(g(x))$, então existem constantes C_1 e C_2 com $C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$. Segue que $|f(x)| \leq C_2|g(x)|$ e $|g(x)| \leq (1/C_1)|f(x)|$ para $x > k$. Assim, $f(x)$ é $O(g(x))$ e $g(x)$ é $O(f(x))$. Reciprocamente, suponha que $f(x)$ seja $O(g(x))$ e que $g(x)$ seja $O(f(x))$. Então, existem constantes C_1, C_2, k_1 e k_2 tais que $|f(x)| \leq C_1|g(x)|$ para $x > k_1$ e $|g(x)| \leq C_2|f(x)|$ para $x > k_2$. Podemos supor que $C_2 > 0$ (podemos sempre aumentar C_2). Então, temos $(1/C_2)|g(x)| \leq |f(x)| \leq C_1|g(x)|$ para $x > \max(k_1, k_2)$. Logo, $f(x)$ é $\Theta(g(x))$. **27.** Se $f(x)$ for $\Theta(g(x))$, então $f(x)$ será tanto $O(g(x))$ quanto $\Omega(g(x))$. Portanto, existem constantes positivas C_1, k_1, C_2 e k_2 tais que $|f(x)| \leq C_2|g(x)|$ para todo $x > k_2$ e $|f(x)| \geq C_1|g(x)|$ para todo $x > k_1$. Segue que $C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$ sempre que $x > k$, em que $k = \max(k_1, k_2)$. Reciprocamente, se existirem constantes positivas C_1, C_2 e k tais que $C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$ para $x > k$, então considerar $k_1 = k_2 = k$ mostra que $f(x)$ é tanto $O(g(x))$ quanto $\Theta(g(x))$.

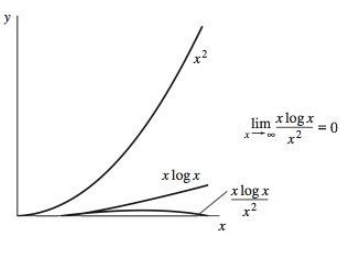
29.



31. Se $f(x)$ for $\Theta(1)$, então $|f(x)|$será limitada por constantes positivas C_1 e C_2. Em outras palavras, $f(x)$ não pode crescer mais do que um limitante fixo ou se tornar menor do que o oposto deste limitante e não deve se tornar mais próximo de 0 do que algum limitante fixo. **33.** Como $f(x)$ é $O(g(x))$, existem constantes C e l tais que $|f(x)| \leq C|g(x)|$ para $x > l$. Logo, $|f^k(x)| \leq C^k|g^k(x)|$ para $x > l$, de modo que $f^k(x)$ é $O(g^k(x))$ considerando a constante como C^k. **35.** Como $f(x)$ e $g(x)$ são crescentes e ilimitadas, podemos supor que $f(x) \geq 1$ e $g(x) \geq 1$ para x suficientemente grande. Existem constantes C e k com $f(x) \leq Cg(x)$ para $x > k$. Isto implica que $\log f(x) \leq \log C + \log g(x) < 2 \log g(x)$ para x suficientemente grande. Logo, $\log f(x)$ é $O(\log g(x))$. **37.** Por definição, existem constantes positivas C_1, C_1', C_2, C_2', k_1, k_1', k_2 e k_2' tais que $f_1(x) \geq C_1|g(x)|$ para todo $x > k_1$, $f_1(x) \leq C_1'|g(x)|$ para todo $x > k_1'$, $f_2(x) \geq C_2|g(x)|$ para todo $x > k_2$ e $f_2(x) \leq C_2'|g(x)|$ para todo $x > k_2'$. A adição da primeira e terceira desigualdades mostra que $f_1(x) + f_2(x) \geq (C_1 + C_2)|g(x)|$ para todo $x > k$ em que $k = \max(k_1, k_2)$. A adição da segunda e quarta desigualdades mostra que $f_1(x) + f_2(x) \leq (C_1' + C_2')|g(x)|$ para todo $x > k'$em que

$k' = \max(k_1', k_2')$. Logo, $f_1(x) + f_2(x)$ é $\Theta(g(x))$. Isto não é mais verdade se f_1 e f_2 puderem assumir valores negativos. **39.** Isto é falso. Sejam $f_1 = x^2 + 2x$, $f_2(x) = x^2 + x$ e $g(x) = x^2$. Então, $f_1(x)$ e $f_2(x)$ são ambas $O(g(x))$, mas $(f_1 - f_2)(x)$ não é. **41.** Considere $f(n)$ como a função com $f(n) = n$ se n for um inteiro positivo ímpar e $f(n) = 1$ se n for um inteiro positivo par e $g(n)$ como a função com $g(n) = 1$ se n for um inteiro positivo ímpar e $g(n) = n$ se n for um inteiro positivo par. **43.** Existem constantes positivas C_1, C_2, C_1', C_2', k_1, k_1', k_2 e k_2' tais que $|f_1(x)| \geq C_1|g_1(x)|$ para todo $x > k_1$, $|f_1(x)| \leq C_1'|g_1(x)|$ para todo $x \geq k_1'$, $|f_2(x)| > C_2|g_2(x)|$ para todo $x > k_2$ e $|f_2(x)| \leq C_2'|g_2(x)|$ para todo $x > k_2'$. Como f_2 e g_2 nunca são nulas, as duas últimas desigualdades podem ser reescritas como $|1/f_2(x)| \leq (1/C_2)|1/g_2(x)|$ para todo $x > k_2$ e $|1/f_2(x)| \geq (1/C_2')|1/g_2(x)|$ para todo $x > k_2'$. Multiplicando a primeira e reescrevendo a quarta desigualdade mostra que $|f_1(x)/f_2(x)| \geq (C_1/C_2')|g_1(x)/g_2(x)|$ para todo $x > \max(k_1, k_2')$, e multiplicando a segunda e reescrevendo a terceira desigualdade fornece $|f_1(x)/f_2(x)| \leq (C_1'/C_2)|g_1(x)/g_2(x)|$ para todo $x > \max(k_1', k_2)$. Segue que f_1/f_2 é Teta grande de g_1/g_2. **45.** Existem constantes positivas C_1, C_2, k_1, k_2, k_1', k_2' tais que $|f(x, y)| \leq C_1|g(x, y)|$ para todo $x > k_1$ e $y > k_2$ e $|f(x, y)| \geq C_2|g(x, y)|$ para todo $x > k_1'$ e $y > k_2'$. **47.** $(x^2 + xy + x \log y)^3 < (3x^2y^3) = 27x^6y^3$ para $x > 1$ e $y > 1$, portanto $x^2 < x^2y$, $xy < x^2y$ e $x \log y < x^2y$. Logo, $(x^2 + xy + x \log y)^3$ é $O(x^6y^3)$. **49.** Para todos os números reais positivos x e y, $\lfloor xy \rfloor \leq xy$. Logo, $\lfloor xy \rfloor$ é $O(xy)$ a partir da definição, considerando $C = 1$ e $k_1 = k_2 = 0$. **51. a)** $\lim_{x\to\infty} x^2/x^3 = \lim_{x\to\infty} 1/x = 0$ **b)** $\lim_{x\to\infty} \frac{x \log x}{x^2} = \lim_{x\to\infty} \frac{\log x}{x} = \lim_{x\to\infty} \frac{1}{x \ln 2} = 0$ (usando a regra de L'Hôpital) **c)** $\lim_{x\to\infty} \frac{x^2}{2^x} = \lim_{x\to\infty} \frac{2x}{2^x \cdot \ln 2} = \lim_{x\to\infty} \frac{2}{2^x \cdot (\ln 2)^2} = 0$ (usando a regra de L'Hôpital) **d)** $\lim_{x\to\infty} \frac{x^2 + x + 1}{x^2} = \lim_{x\to\infty} (1 + \frac{1}{x} + \frac{1}{x^2}) = 1 \neq 0$

53.



55. Não. Considere $f(x) = 1/x^2$ e $g(x) = 1/x$. **57. a)** Como $\lim_{x\to\infty} f(x)/g(x) = 0$, $|f(x)|/|g(x)| < 1$ para x suficientemente grande . Logo, $|f(x)| < |g(x)|$ para $x > k$ para alguma constante k. Portanto, $f(x)$ é $O(g(x))$. **b)** Seja $f(x) = g(x) = x$. Então, $f(x)$ é $O(g(x))$, mas $f(x)$ não é $o(g(x))$ porque $f(x)/g(x) = 1$. **59.** Como $f_2(x)$ é $o(g(x))$, do Exercício 57(a) segue que $f_2(x)$ é $O(g(x))$. Pelo Corolário 1, temos que $f_1(x) + f_2(x)$ é $O(g(x))$. **61.** Podemos mostrar facilmente que $(n - i)(i + 1) \geq n$ para $i = 0, 1, \ldots, n - 1$. Logo, $(n!)^2 = (n \cdot 1)((n - 1) \cdot 2) \cdot ((n - 2) \cdot 3) \cdots (2 \cdot (n - 1)) \cdot (1 \cdot n) \geq n^n$. Portanto, $2 \log n! \geq n \log n$. **63.** Calcule que $\log 5! \approx 6{,}9$ e $(5 \log 5)/4 \approx 2{,}9$, de modo que a desigualdade vale para $n = 5$. Suponha que $n \geq 6$. Como $n!$ é o produto de todos os inteiros de n até 1, temos $n! > n(n-1)(n-2) \cdots \lceil n/2 \rceil$ (pois pelo menos o termo 2 está faltando). Observe que existem mais de $n/2$ termos neste produto, e cada termo é pelo menos tão grande quanto $n/2$. Portanto, o produto é maior que $(n/2)^{(n/2)}$. Considerando o log de ambos os lados da desigualdade, temos $\log n! > \log (\frac{n}{2})^{n/2} = \frac{n}{2} \log \frac{n}{2} = \frac{n}{2}(\log n \cdot -1) > (n \log n)/4$, pois $n > 4$ implica que $\log n - 1 > (\log n)/2$. **65.** Todas são não assintóticas.

Seção 3.3

1. $2n - 1$ **3.** Linear **5.** $O(n)$ **7. a)** *potencia* := 1, $y := 1$; $i := 1$, *potencia* := 2, $y := 3$; $i := 2$, *potencia* := 4, $y := 15$ **b)** $2n$ multiplicações e n adições **9. a)** $2^{10^9} \approx 10^{3 \times 10^8}$ **b)** 10^9 **c)** $3{,}96 \times 10^7$ **d)** $3{,}16 \times 10^4$ **e)** 29 **f)** 12 **11. a)** 36 anos **b)** 13 dias **c)** 19 minutos **13.** O número médio de comparações é $(3n + 4)/2$. **15.** $O(\log n)$ **17.** $O(n)$ **19.** $O(n^2)$ **21.** $O(n)$ **23.** $O(n)$ **25.** $O(\log n)$ comparações; $O(n^2)$ trocas **27. a)** dobra **b)** aumenta de 1

Seção 3.4

1. a) Sim. **b)** Não. **c)** Sim. **d)** Não. **3.** Suponha que $a \mid b$. Então, existe um inteiro k tal que $ka = b$. Como $a(ck) = bc$, segue que $a \mid bc$. **5.** Se $a \mid b$ e $b \mid a$, existem inteiros c e d tais que $b = ac$ e $a = bd$. Logo, $a = acd$. Como $a \neq 0$, segue que $cd = 1$. Assim, ou $c = d = 1$ ou $c = d = -1$. Logo, ou $a = b$ ou $a = -b$. **7.** Como $ac \mid bc$, existe um inteiro k tal que $ack = bc$. Logo, $ak = b$, de modo que $a \mid b$. **9. a)** 2, 5 **b)** -11, 10 **c)** 34, 7 **d)** 77, 0 **e)** 0, 0 **f)** 0, 3 **g)** $-1{,}2$ **h)** 4, 0 **11.** Se a **mod** $m = b$ **mod** m, então a e b têm o mesmo resto quando divididos por m. Logo, $a = q_1m + r$ e $b = q_2m + r$, em que $0 \leq r < m$. Segue que $a - b = (q_1 - q_2)m$, de modo que $m \mid (a - b)$. Segue que $a \equiv b \pmod{m}$. **13.** Existe algum b com $(b-1)\,k < n \leq bk$. Logo, $(b-1)\,k \leq n - 1 < bk$. Divida por k para obter $b - 1 < n/k \leq b$ e $b - 1 \leq (n-1)/k < b$. Logo, $\lceil n/k \rceil = b$ e $\lfloor (n-1)/k \rfloor = b - 1$. **15.** x **mod** m se x **mod** $m \leq \lceil m/2 \rceil$ e $(x$ **mod** $m) - m$ se x **mod** $m > \lceil m/2 \rceil$ **17. a)** 1 **b)** 2 **c)** 3 **d)** 9 **19. a)** Não. **b)** Não. **c)** Sim. **d)** Não. **21.** Seja $m = tn$. Como $a \equiv b \pmod{m}$ existe um inteiro s tal que $a = b + stn$. Logo, $a = b + (st)n$, de modo que $a \equiv b \pmod{n}$. **23. a)** Sejam $m = c = 2$, $a = 0$ e $b = 1$. Então, $0 = ac \equiv bc = 2 \pmod 2$, mas $0 = a \not\equiv b = 1 \pmod 2$. **b)** Sejam $m = 5$, $a = b = 3$, $c = 1$ e $d = 6$. Então, $3 \equiv 3 \pmod 5$ e $1 \equiv 6 \pmod 5$, mas $3^1 = 3 \not\equiv 4 \equiv 729 = 3^6 \pmod 5$. **25.** Como $a \equiv b \pmod{m}$, existe um inteiro s tal que $a = b + sm$, de modo que $a - b = sm$. Então, $a^k - b^k = (a - b)(a^{k-1} + a^{k-2}b + \cdots + ab^{k-2} + b^{k-1})$, $k \geq 2$, também é um múltiplo de m. Segue que $a^k \equiv b^k \pmod{m}$. **27. a)** 7, 19, 7, 7, 18, 0 **b)** Considere o próximo espaço disponível **mod** 31. **29.** 2, 6, 7, 10, 8, 2, 6, 7, 10, 8, ... **31. a)** GR QRW SD VV JR **b)** QB ABG CNFF TB **c)** QX UXM AHJJ ZX. **33.** 4. **35.** O algarismo de verifi-

cação do ISBN para este livro é válido porque $1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8 + 10 \cdot 2 \equiv 0 \pmod{11}$.

Seção 3.5

1. 29, 71, 97 primos; 21, 111, 143 não primos **3. a)** $2^3 \cdot 11$ **b)** $2 \cdot 3^2 \cdot 7$ **c)** 3^6 **d)** $7 \cdot 11 \cdot 13$ **e)** $11 \cdot 101$ **f)** $2 \cdot 3^3 \cdot 5 \cdot 7 \cdot 13 \cdot 37$ **5.** $2^8 \cdot 3^4 \cdot 5^2 \cdot 7$ **7.** Suponha que $\log_2 3 = a/b$, em que $a, b \in \mathbf{Z}^+$ e $b \neq 0$. Então $2^{a/b} = 3$, de modo que $2^a = 3^b$. Isto viola o Teorema Fundamental da Aritmética. Logo, $\log_2 3$ é irracional. **9.** 3, 5 e 7 são primos da forma desejada. **11.** 1, 7, 11, 13, 17, 19, 23, 29 **13. a)** Sim. **b)** Não. **c)** Sim. **d)** Sim. **15.** Suponha que n não seja primo, de modo que $n = ab$, em que a e b são inteiros maiores que 1. Como $a > 1$, pela identidade dada na dica, $2^a - 1$ é um fator de $2^n - 1$ que é maior que 1, e o segundo fator nesta identidade também é maior que 1. Logo, $2^n - 1$ não é primo. **17. a)** 2 **b)** 4 **c)** 12 **19.** $\phi(p^k) = p^k - p^{k-1}$ **21. a)** $3^5 \cdot 5^3$ **b)** 1 **c)** 23^{17} **d)** $41 \cdot 43 \cdot 53$ **e)** 1 **f)** 1111 **23. a)** $2^{11} \cdot 3^7 \cdot 5^9 \cdot 7^3$ **b)** $2^9 \cdot 3^7 \cdot 5^5 \cdot 7^3 \cdot 11 \cdot 13 \cdot 17$ **c)** 23^{31} **d)** $41 \cdot 43 \cdot 53$ **e)** $2^{12}3^{13}5^{17}7^{21}$ **f)** Não definido. **25.** mdc (92928, 123552) = 1056; mmc(92928, 123552) = 10 872 576; ambos os produtos são 11 481 440 256. **27.** Como $\min(x, y) + \max(x, y) = x + y$, o expoente de p_i na fatoração em primos de $\text{mdc}(a, b) \cdot \text{mmc}(a, b)$ é a soma dos expoentes de p_i nas fatorações em primos de a e b. **29. a)** $a_n = 1$ se n é primo e $a_n = 0$ caso contrário. **b)** a_n é o menor fator primo de n com $a_1 = 1$. **c)** a_n é o número de divisores positivos de n. **d)** $a_n = 1$ se n não tiver nenhum divisor que seja um quadrado perfeito maior que 1 e $a_n = 0$ caso contrário. **e)** a_n é o maior primo menor que ou igual a n. **f)** a_n é o produto dos primeiros $n-1$ primos. **31.** Como um a cada dois inteiros é divisível por 2, o produto é divisível por 2. Como um a cada três inteiros é divisível por 3, o produto é divisível por 3. Portanto, o produto tem ambos, 2 e 3, em sua fatoração em primos e é, portanto, divisível por $3 \cdot 2 = 6$. **33.** $n = 1601$ é um contra-exemplo. **35.** Suponha que exista apenas um número finito de primos da forma $4k + 3$, a saber, $q_1, q_2, \ldots, q_n$, em que $q_1 = 3, q_2 = 7$, e assim por diante. Seja $Q = 4q_1q_2 \cdots q_n - 1$. Observe que Q é da forma $4k + 3$ (em que $k = q_1q_2 \cdots q_n - 1$). Se Q for primo, então encontramos um primo da forma desejada diferente de todos os listados. Se Q não for primo, então Q tem pelo menos um fator primo que não está na lista $q_1, q_2, \ldots, q_n$, porque o resto, quando Q é dividido por q_j, é $q_j - 1$, e $q_j - 1 \neq 0$. Como todos os primos ímpares são ou da forma $4k + 1$ ou da forma $4k + 3$, e o produto de primos da forma $4k + 1$ também é desta forma (porque $(4k + 1)(4m + 1) = 4(4km + k + m) + 1$), deve existir um fator de Q da forma $4k + 3$ diferente dos primos listados. **37.** Dado um inteiro positivo x, mostramos que existe exatamente um número racional positivo m/n (em fatores primos entre si) tal que $K(m/n) = x$. Da fatoração em primos de x, escolha m e n tais que $K(m/n) = x$. Os primos que ocorrem em potências pares são os primos que ocorrem na fatoração em primos de m, com os expoentes sendo a metade dos expoentes correspondentes em x; e os primos que ocorrem em potências ímpares são os primos que ocorrem na fatoração em primos de n, com os expoentes sendo a metade de um a mais que os expoentes em x.

Seção 3.6

1. a) 1110 0111 **b)** 1 0001 1011 0100 **c)** 1 0111 1101 0110 1100 **3. a)** 31 **b)** 513 **c)** 341 **d)** 26 896 **5. a)** 1000 0000 1110 **b)** 1 0011 0101 1010 1011 **c)** 1010 1011 1011 1010 **d)** 1101 1110 1111 1010 11001110 1101 **7.** 1010 1011 1100 1101 1110 1111 **9.** $(\text{B7B})_{16}$ **11.** Adicionando até três 0s dominantes, se necessário, escreva a expansão binária como $(\ldots b_{23}b_{22}b_{21}b_{20}b_{13}b_{12}b_{11}b_{10}b_{03}b_{02}b_{01}b_{00})_2$. O valor deste numeral é $b_{00} + 2b_{01} + 4b_{02} + 8b_{03} + 2^4b_{10} + 2^5b_{11} + 2^6b_{12} + 2^7b_{13} + 2^8b_{20} + 2^9b_{21} + 2^{10}b_{22} + 2^{11}b_{23} + \cdots$, o qual podemos reescrever como $b_{00} + 2b_{01} + 4b_{02} + 8b_{03} + (b_{10} + 2b_{11} + 4b_{12} + 8b_{13}) \cdot 2^4 + (b_{20} + 2b_{21} + 4b_{22} + 8b_{23}) \cdot 2^8 + \cdots$. Agora, $(b_{i3}b_{i2}b_{i1}b_{i0})_2$ se traduz no dígito hexadecimal h_i. Assim, nosso número é $h_0 + h_1 \cdot 2^4 + h_2 \cdot 2^8 + \cdots = h_0 + h_1 \cdot 16 + h_2 \cdot 16^2 + \cdots$, que é a expansão hexadecimal $(\ldots h_1h_1h_0)_{16}$. **13.** Agrupe blocos de três dígitos binários, adicionando até dois 0s iniciais, se necessário, e traduza cada bloco de três dígitos binários em um único algarismo octal. **15.** $(1110111001010111010001)_2$, $(1273)_8$ **17.** Converta o número octal dado para binário usando o Exercício 14, então converta de binário para hexadecimal usando o Exemplo 6. **19.** 436 **21.** 27 **23. a)** 6 **b)** 3 **c)** 11 **d)** 3 **e)** 40 **f)** 12 **25.** 8 **27.** A expansão binária do inteiro é a única destas somas. **29.** Seja $a = (a_{n-1}a_{n-2} \ldots a_1a_0)_{10}$. Então, $a = 10^{n-1}a_{n-1} + 10^{n-2}a_{n-2} + \cdots + 10a_1 + a_0 \equiv a_{n-1} + a_{n-2} + \cdots + a_1 + a_0 \pmod 3$, porque $10^j \equiv 1 \pmod 3$ para todo inteiro não negativo j. Segue que $3 \mid a$ se e somente se 3 divide a soma dos dígitos decimais de a. **31.** Seja $a = (a_{n-1}a_{n-2}\cdots a_1a_0)_2$. Então, $a = a_0 + 2a_1 + 2^2a_2 + \cdots + 2^{n-1}a_{n-1} \equiv a_0 - a_1 + a_2 - a_3 + \cdots \pm a_{n-1} \pmod 3$. Segue que a é divisível por 3 se e somente se a soma dos dígitos binários nas posições com numeração par menos a soma dos dígitos binários nas posições com numeração ímpar é divisível por 3. **33. a)** -6 **b)** 13 **c)** -14 **d)** 0 **35.** O complemento de um da soma é encontrado somando os complementos de um dos dois inteiros, exceto que um "vai um" no bit dominante é usado para armazenar o último bit da soma. **37.** Se $m \geq 0$, então o bit dominante a_{n-1} da expansão em complemento de um de m é 0 e a fórmula fica $m = \sum_{i=0}^{n-2} a_i 2^i$. Isto é correto, pois o lado direito é a expansão binária de m. Quando m é negativo, o bit dominante a_{n-1} da expansão em complementos de um de m é 1. Os $n - 1$ bits restantes podem ser obtidos subtraindo $-m$ de 111 ... 1 (onde existem $n - 1$ 1s), pois subtrair um bit de 1 é o mesmo que considerar seu complementar. Logo, a seqüência de bits $a_{n-2} \ldots a_0$ é a expansão binária de $(2^{n-1} - 1) - (-m)$. Isolando m na equação $(2^{n-1} - 1) - (-m) = \sum_{i=0}^{n-2} a_i 2^i$ fornece a equação desejada, pois $a_{n-1} = 1$. **39. a)** -7 **b)** 13 **c)** -15 **d)** -1 **41.** Para obter a representação em complemento de dois da

soma de dois inteiros, some suas representações em complemento de dois (do modo como inteiros binários são somados) e ignore qualquer "vai um" da coluna mais à esquerda. Entretanto, a resposta é inválida se ocorrer um *overflow*. Isto ocorre quando os dígitos mais à esquerda nas duas representações em complemento de dois dos dois termos coincidem e o algarismo mais à esquerda da resposta for diferente. **43.** Se $m \geq 0$, então o bit dominante a_{n-1} é 0 e a fórmula fica $m = \sum_{i=0}^{n-2} a_i 2^i$. Isto é correto porque o lado direito é a expansão binária de m. Se $m < 0$, sua expansão em complemento de dois tem 1 como seu bit dominante e os $n-1$ bits restantes são a expansão binária de $2^{n-1} - (-m)$. Isto significa que $(2^{n-1}) - (-m) = \sum_{i=0}^{n-2} a_i 2^i$. Isolando m obtém-se a equação desejada, porque $a_{n-1} = 1$. **45.** $4n$

47. **procedure** *Cantor*(x: inteiro positivo)
$n := 1; f := 1$
while $(n+1) \cdot f \leq x$
begin
$n := n + 1$
$f := f \cdot n$
end
$y := x$
while $n > 0$
begin
$a_n := \lfloor y/f \rfloor$
$y := y - a_n \cdot f$
$f := f/n$
$n := n - 1$
end $\{x = a_n n! + a_{n-1}(n-1)! + \cdots + a_1 1!\}$

49. Primeiro passo: $c = 0$, $d = 0$, $s_0 = 1$; segundo passo: $c = 0$, $d = 1$, $s_1 = 0$; terceiro passo: $c = 1$, $d = 1$, $s_2 = 0$; quarto passo: $c = 1$, $d = 1$, $s_3 = 0$; quinto passo: $c = 1$, $d = 1$, $s_4 = 1$; sexto passo: $c = 1$, $s_5 = 1$

51. **procedure** *subtrai*(a, b: inteiros positivos, $a > b$,
$a = (a_{n-1}a_{n-2} \ldots a_1 a_0)_2$,
$b = (b_{n-1}b_{n-2} \ldots b_1 b_0)_2$)
$B := 0$ {B é o empréstimo}
for $j := 0$ **to** $n-1$
begin
se $a_j \geq b_j + B$ **then**
begin
$s_j := a_j - b_j - B$
$B := 0$
end
else
begin
$s_j := a_j + 2 - b_j - B$
$B := 1$
end
end $\{(s_{n-1}s_{n-2} \ldots s_1 s_0)_2$ é a diferença}

53. **procedure** *compara*(a, b: inteiros positivos,
$a = (a_n a_{n-1} \ldots a_1 a_0)_2$,
$b = (b_n b_{n-1} \ldots b_1 b_0)_2$)
$k := n$
while $a_k = b_k$ e $k > 0$
$k := k - 1$
if $a_k = b_k$ **then** imprima "a é igual a b"
if $a_k > b_k$ **then** imprima "a é maior que b"
if $a_k < b_k$ **then** imprima "a é menor que b"

55. $O(\log n)$ **57.** A única parte do algoritmo que consome tempo é o laço **while**, que é iterado q vezes. O trabalho feito dentro é a subtração de inteiros que não sejam maiores que a, que tem log a bits. O resultado agora segue do Exemplo 8.

Seção 3.7

1. **a)** $1 = (-1) \cdot 10 + 1 \cdot 11$ **b)** $1 = 21 \cdot 21 + (-10) \cdot 44$ **c)** $12 = (-1) \cdot 36 + 48$ **d)** $1 = 13 \cdot 55 + (-21) \cdot 34$ **e)** $3 = 11 \cdot 213 + (-20) \cdot 117$ **f)** $223 = 1 \cdot 0 + 1 \cdot 223$ **g)** $1 = 37 \cdot 2347 + (-706) \cdot 123$ **h)** $2 = 1128 \cdot 3454 + (-835) \cdot 4666$ **i)** $1 = 2468 \cdot 9999 + (-2221) \cdot 11111$ **3.** $15 \cdot 7 = 105 \equiv 1 \pmod{26}$ **5.** 7 **7.** 52 **9.** Suponha que b e c sejam ambos inversos de a módulo m. Então $ba \equiv 1 \pmod{m}$ e $ca \equiv 1 \pmod{m}$. Logo, $ba \equiv ca \pmod{m}$. Como $\text{mdc}(a, m) = 1$, segue do Teorema 2 que $b \equiv c \pmod{m}$. **11.** $x \equiv 8 \pmod{9}$ **13.** Seja $m' = m/\text{mdc}(c, m)$. Como todos os fatores comuns de m e c foram cancelados de m para obter m', segue que m' e c são primos entre si. Como m divide $(ac - bc) = (a - b)c$, segue que m' divide $(a - b)c$. Pelo Lema 1, vemos que m' divide $a - b$, de modo que $a \equiv b \pmod{m'}$. **15.** Suponha que $x^2 \equiv 1 \pmod{p}$. Então p divide $x^2 - 1 = (x+1)(x-1)$. Pelo Lema 2, segue que $p \mid (x+1)$ ou $p \mid (x-1)$, de modo que $x \equiv -1 \pmod{p}$ ou $x \equiv 1 \pmod{p}$. **17.** **a)** Suponha que $ia \equiv ja \pmod{p}$, em que $1 \leq i < j < p$. Então p divide $ja - ia = a(j - i)$. Pelo Teorema 1, como a não é divisível por p, p divide $j - i$, o que é impossível, pois $j - i$ é um inteiro positivo menor que p. **b)** Pela parte (a), como nenhum par dentre $a, 2a, \ldots, (p-1)a$ é congruente módulo p, cada um deve ser congruente a um número diferente de 1 para $p - 1$. Segue que $a \cdot 2a \cdot 3a \cdot \cdots \cdot (p-1) \cdot a \equiv 1 \cdot 2 \cdot 3 \cdot \cdots \cdot (p-1) \pmod{p}$. Segue que $(p-1)! \cdot a^{p-1} \equiv (p-1)! \pmod{p}$. **c)** Pelo Teorema de Wilson e pela parte (b), se p não dividir a, segue que $(-1) \cdot a^{p-1} \equiv -1 \pmod{p}$. Logo, $a^{p-1} \equiv 1 \pmod{p}$. **d)** Se $p \mid a$, então $p \mid a^p$. Logo, $a^p \equiv a \equiv 0 \pmod{p}$. Se p não dividir a, então $a^{p-1} \equiv a \pmod{p}$, pela parte (c). Multiplicar ambos os lados desta congruência por a fornece $a^p \equiv a \pmod{p}$. **19.** Todos os inteiros da forma $323 + 330k$, em que k é um inteiro. **21.** Todos os inteiros da forma $16 + 252k$, em que k é um inteiro. **23.** Suponha que p seja um primo que apareça na fatoração em primos de $m_1 m_2 \cdots m_n$. Como os m_i s são primos entre si, p é um fator de exatamente um dos m_i s, digamos m_j. Como m_j divide $a - b$, segue que $a - b$ tem o fator p em sua fatoração em primos elevado a uma potência pelo menos tão grande quanto a potência com a qual ele aparece na fatoração em primos de m_j. Segue que $m_1 m_2 \cdots m_n$ divide $a - b$, de modo que $a \equiv b \pmod{m_1 m_2 \cdots m_n}$. **25.** $x \equiv 1 \pmod{6}$ **27.** **a)** Pelo Pequeno Teorema de Fermat, temos $2^{10} \equiv 1 \pmod{11}$. Logo, $2^{340} = (2^{10})^{34} \equiv 1^{34} = 1 \pmod{11}$. **b)** Como $32 \equiv 1 \pmod{31}$, segue que 2^{340}

$= (2^5)^{68} = 32^{68} \equiv 1^{68} = 1 \pmod{31}$. **c)** Como 11 e 31 são primos entre si, e $11 \cdot 31 = 341$, segue das partes (a) e (b) e do Exercício 23 que $2^{340} \equiv 1 \pmod{341}$. **29. a)** 3, 4, 8 **b)** 983 **31.** Primeiro, $2047 = 23 \cdot 89$ é composto. Escreva $2047 - 1 = 2046 = 2 \cdot 1023$, de modo que $s = 1$ e $t = 1023$ na definição. Então, $2^{1023} = (2^{11})^{93} = 2048^{93} \equiv 1^{93} = 1 \pmod{2047}$, como desejado. **33.** Devemos mostrar que $b^{2820} \equiv 1 \pmod{2821}$ para todo b relativamente primo a 2821. Observe que $2821 = 7 \cdot 13 \cdot 31$, e se $\text{mdc}(b, 2821) = 1$, então $\text{mdc}(b, 7) = \text{mdc}(b, 13) = \text{mdc}(b, 31) = 1$. Usando o Pequeno Teorema de Fermat, descobrimos que $b^6 \equiv 1 \pmod 7$, $b^{12} \equiv 1 \pmod{13}$ e $b^{30} \equiv 1 \pmod{31}$. Segue que $b^{2820} \equiv (b^6)^{470} \equiv 1 \pmod 7$, $b^{2820} \equiv (b^{12})^{235} \equiv 1 \pmod{13}$ e $b^{2820} \equiv (b^{30})^{94} \equiv 1 \pmod{31}$. Pelo Exercício 23 (ou pelo Teorema Chinês do Resto) segue que $b^{2820} \equiv 1 \pmod{2821}$, como desejado. **35. a)** Se cancelarmos esta expressão, obtemos $n = 1296m^3 + 396m^2 + 36m + 1$. Claramente, $6m \mid n - 1$, $12m \mid n - 1$ e $18m \mid n - 1$. Portanto, as condições do Exercício 34 são satisfeitas, e concluímos que n é um número de Carmichael. **b)** Considerando $m = 51$, obtemos $n = 172\,947\,529$. **37.** $0 = (0, 0)$, $1 = (1, 1)$, $2 = (2, 2)$, $3 = (0, 3)$, $4 = (1, 4)$, $5 = (2, 0)$, $6 = (0, 1)$, $7 = (1, 2)$, $8 = (2, 3)$, $9 = (0, 4)$, $10 = (1, 0)$, $11 = (2, 1)$, $12 = (0, 2)$, $13 = (1, 3)$, $14 = (2, 4)$ **39.** Temos $m_1 = 99$, $m_2 = 98$, $m_3 = 97$ e $m_4 = 95$, de modo que $m = 99 \cdot 98 \cdot 97 \cdot 95 = 89\,403\,930$. Encontramos que $M_1 = m/m_1 = 903\,070$, $M_2 = m/m_2 = 912\,285$, $M_3 = m/m_3 = 921\,690$ e $M_4 = m/m_4 = 941\,094$. Usando o Algoritmo de Euclides, calculamos que $y_1 = 37$, $y_2 = 33$, $y_3 = 24$ e $y_4 = 4$ são inversos de M_k módulo m_k para $k = 1, 2, 3, 4$, respectivamente. Segue que a solução é $65 \cdot 903\,070 \cdot 37 + 2 \cdot 912\,285 \cdot 33 + 51 \cdot 921\,690 \cdot 24 + 10 \cdot 941\,094 \cdot 4 = 3\,397\,886\,480 \equiv 537\,140 \pmod{89\,403\,930}$. **41.** Pelo Exercício 40, segue que $\text{mdc}(2^b - 1, (2^a - 1) \bmod (2^b - 1)) = \text{mdc}(2^b - 1, 2^{a \bmod b} - 1)$. Como os expoentes envolvidos no cálculo são b e $a \bmod b$, os mesmos que as quantidades envolvidas no cálculo de $\text{mdc}(a, b)$, os passos usados no algoritmo de Euclides para calcular $\text{mdc}(2^a - 1, 2^b - 1)$ são paralelos àqueles usados para calcular $\text{mdc}(a, b)$ e mostram que $\text{mdc}(2^a - 1, 2^b - 1) = 2^{\text{mdc}(a, b)} - 1$. **43.** Suponha que q seja um primo ímpar com $q \mid 2^p - 1$. Pelo PequenoTeorema de Fermat, $q \mid 2^{q-1} - 1$. Do Exercício 41, $\text{mdc}(2^p - 1, 2^{q-1} - 1) = 2^{\text{mdc}(p, q-1)} - 1$. Como q é um divisor comum de $2^p - 1$ e $2^{q-1} - 1$, $\text{mdc}(2^p - 1, 2^{q-1} - 1) > 1$. Logo, $\text{mdc}(p, q - 1) = p$, pois a única outra possibilidade, a saber, $\text{mdc}(p, q - 1) = 1$, nos dá $\text{mdc}(2^p - 1, 2^{q-1} - 1) = 1$. Logo, $p \mid q - 1$, e portanto existe um inteiro positivo m tal que $q - 1 = mp$. Como q é ímpar, m deve ser par, digamos, $m = 2k$, e assim todo divisor primo de $2^p - 1$ é da forma $2kp + 1$. Além disso, o produto de números desta forma também é desta forma. Portanto, todos os divisores de $2^p - 1$ são desta forma. **45.** Suponha que saibamos que $n = pq$ e $(p - 1)(q - 1)$. Para encontrar p e q, observamos primeiro que $(p - 1)(q - 1) = pq - p - q + 1 = n - (p + q) + 1$. Disto, podemos encontrar $s = (p + q)$. Como $q = s - p$, temos $n = p(s - p)$. Logo, $p^2 - ps + n = 0$. Agora, podemos usar a fórmula quadrática para encontrar p. Uma vez que tenhamos encontrado p, podemos encontrar q, pois $q = n/p$. **47.** SILVER **49.** $34 \cdot 144 + (-55) \cdot 89 = 1$

51. procedure *Euclides estendido* (a, b: inteiros positivos)

$x := a$
$y := b$
$oldolds := 1$
$olds := 0$
$oldoldt := 0$
$oldt := 1$
while $y \neq 0$
begin
  $q := x$ **div** y
  $r := x$ **mod** y
  $x := y$
  $y := r$
  $s := oldolds - q \cdot olds$
  $t := oldoldt - q \cdot oldt$
  $oldolds := olds$
  $oldoldt := oldt$
  $olds := s$
  $oldt := t$
end {$\text{mdc}(a, b)$ é x e $(oldolds)a + (oldoldt)b = x$}

53. Suponha que s seja uma solução de $x^2 \equiv a \pmod p$. Então, como $(-s)^2 = s^2$, $-s$ também é uma solução. Além disso, $s \not\equiv -s \pmod p$. Caso contrário, $p \mid 2s$, o que implica que $p \mid s$, e isto implica, usando a hipótese original, que $p \mid a$, o que é uma contradição. Além disso, se s e t são soluções não congruentes módulo p, então, como $s^2 \equiv t^2 \pmod p$, $p \mid (s^2 - t^2)$. Isto implica que $p \mid (s + t)(s - t)$, e pelo Lema 2, $p \mid (s - t)$ ou $p \mid (s + t)$, de modo que $s \equiv t \pmod p$ ou $s \equiv -t \pmod p$. Logo, existem no máximo duas soluções. **55.** O valor de $\left(\frac{a}{p}\right)$ depende apenas de a ser um resíduo quadrático módulo p, isto é, de $x^2 \equiv a \pmod p$ ter uma solução. Como isto depende apenas da classe de equivalência de a módulo p, segue que $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$ se $a \equiv b \pmod p$. **57.** Pelo Exercício 56, $\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = a^{(p-1)/2}b^{(p-1)/2} = (ab)^{(p-1)/2} \equiv \left(\frac{ab}{p}\right) \pmod p$. **59.** $x \equiv 8$, 13, 22 ou 27 (mod 35) **61.** Suponha que usemos um primo como n. Para encontrar uma chave de decodificação privada a partir da chave pública de codificação correspondente e, precisaríamos encontrar um número d que seja um inverso de e módulo $n - 1$, de modo que os cálculos mostrados antes do Exemplo 12 possam ser feitos. Mas encontrar um destes d é fácil usando o algoritmo de Euclides, pois a pessoa fazendo isto já conheceria $n - 1$. Em particular, para encontrar d, podemos executar os passos do algoritmo de Euclides de trás para frente para expressar 1 como uma combinação de e e $n - 1$; então d é o coeficiente de e nesta combinação linear.

Seção 3.8

1. a) 3×4 **b)** $\begin{bmatrix} 1 \\ 4 \\ 3 \end{bmatrix}$ **c)** $[2\ 0\ 4\ 6]$ **d)** 1

e) $\begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 1 \\ 1 & 4 & 3 \\ 3 & 6 & 7 \end{bmatrix}$ **3. a)** $\begin{bmatrix} 1 & 11 \\ 2 & 18 \end{bmatrix}$ **b)** $\begin{bmatrix} 2 & -2 & -3 \\ 1 & 0 & 2 \\ 9 & -4 & 4 \end{bmatrix}$

c) $\begin{bmatrix} -4 & 15 & -4 & 1 \\ -3 & 10 & 2 & -3 \\ 0 & 2 & -8 & 6 \\ 1 & -8 & 18 & -13 \end{bmatrix}$ **5.** $\begin{bmatrix} 9/5 & -6/5 \\ -1/5 & 4/5 \end{bmatrix}$

7. $\mathbf{0} + \mathbf{A} = [0 + a_{ij}] = [a_{ij} + 0] = \mathbf{0} + \mathbf{A}$ **9.** $\mathbf{A} + (\mathbf{B} + \mathbf{C}) = [a_{ij} + (b_{ij} + c_{ij})] = [(a_{ij} + b_{ij}) + c_{ij}] = (\mathbf{A} + \mathbf{B}) + \mathbf{C}$ **11.** O número de linhas de **A** é igual ao número de colunas de **B**, e o número de colunas de **A** é igual ao número de linhas de **B**. **13.** $\mathbf{A(BC)} = \left[\sum_q a_{iq}\left(\sum_r b_{qr}c_{rl}\right)\right] = \left[\sum_q \sum_r a_{iq}b_{qr}c_{rl}\right] = \left[\sum_r \left(\sum_q a_{iq}b_{qr}\right)c_{rl}\right] = \mathbf{(AB)C}$ **15.** $\mathbf{A}^n = \begin{bmatrix} 1 & n \\ 0 & 1 \end{bmatrix}$ **17. a)** Seja $\mathbf{A} = [a_{ij}]$ e $\mathbf{B} = [b_{ij}]$. Então, $\mathbf{A} + \mathbf{B} = [a_{ij} + b_{ij}]$. Temos $(\mathbf{A} + \mathbf{B})^t = [a_{ji} + b_{ji}] = [a_{ji}] + [b_{ji}] = \mathbf{A}^t + \mathbf{B}^t$. **b)** Usando a mesma notação da parte (a), temos $\mathbf{B}^t\mathbf{A}^t = \left[\sum_q b_{qi}a_{jq}\right] = \left[\sum_q a_{jq}b_{qi}\right] = (\mathbf{AB})^t$, pois o elemento (i, j) é o elemento (j, i) de **AB**. **19.** O resultado segue porque $\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \begin{bmatrix} ad - bc & 0 \\ 0 & ad - bc \end{bmatrix} = (ad - bc)\mathbf{I}_2 = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix}$. **21.** $\mathbf{A}^n(\mathbf{A}^{-1})^n = \mathbf{A}(\mathbf{A}\cdots(\mathbf{A}(\mathbf{A}\mathbf{A}^{-1})\mathbf{A}^{-1})\cdots\mathbf{A}^{-1})\mathbf{A}^{-1}$ pela propriedade associativa. Como $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$, trabalhando a partir de dentro, $\mathbf{A}^n(\mathbf{A}^{-1})^n = \mathbf{I}$. Analogamente, $(\mathbf{A}^{-1})^n\mathbf{A}^n = \mathbf{I}$. Portanto, $(\mathbf{A}^n)^{-1} = (\mathbf{A}^{-1})^n$. **23.** Existem m_2 multiplicações usadas para encontrar cada elemento m_1m_3 do produto. Logo, são usadas $m_1m_2m_3$ multiplicações. **25.** $\mathbf{A}_1((\mathbf{A}_2\mathbf{A}_3)\mathbf{A}_4)$ **27.** $x_1 = 1$, $x_2 = -1$, $x_3 = -2$

29. a) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ **c)** $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$

31. a) $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ **c)** $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

33. a) $\mathbf{A} \vee \mathbf{B} = [a_{ij} \vee b_{ij}] = [b_{ij} \vee a_{ij}] = \mathbf{B} \vee \mathbf{A}$ **b)** $\mathbf{A} \wedge \mathbf{B} = [a_{ij} \wedge b_{ij}] = [b_{ij} \wedge a_{ij}] = \mathbf{B} \wedge \mathbf{A}$ **35. a)** $\mathbf{A} \vee (\mathbf{B} \wedge \mathbf{C}) = [a_{ij}] \vee [b_{ij} \wedge c_{ij}] = [a_{ij} \vee (b_{ij} \wedge c_{ij})] = [(a_{ij} \vee b_{ij}) \wedge (a_{ij} \vee c_{ij})] = [a_{ij} \vee b_{ij}] \wedge [a_{ij} \vee c_{ij}] = (\mathbf{A} \vee \mathbf{B}) \wedge (\mathbf{A} \vee \mathbf{C})$ **b)** $\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C}) = [a_{ij}] \wedge [b_{ij} \vee c_{ij}] = [a_{ij} \wedge (b_{ij} \vee c_{ij})] = [(a_{ij} \wedge b_{ij}) \vee (a_{ij} \wedge c_{ij})] = [a_{ij} \wedge b_{ij}] \vee [a_{ij} \wedge c_{ij}] = (\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{C})$ **37.** $\mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}) = [\bigvee_q a_{iq} \wedge (\bigvee_r (b_{qr} \wedge c_{rl}))] = [\bigvee_q \bigvee_r (a_{iq} \wedge b_{qr} \wedge c_{rl})] = [\bigvee_r \bigvee_q (a_{iq} \wedge b_{qr} \wedge c_{rl})] = [\bigvee_r (\bigvee_q (a_{iq} \wedge b_{qr})) \wedge c_{rl}] = (\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C}$

Exercícios Complementares

1. a) **procedure** *ultimo max*($a_1, \ldots, a_n$: inteiros)
max := a_1
ultimo := 1
i := 2
while $i \leq n$
begin
if $a_i \geq max$ **then**
begin
max := a_i
ultimo := *i*
end
i := *i* + 1
end {*ultimo* é a posição da última ocorrência do maior inteiro na lista}

b) $2n - 1 = O(n)$ comparações

3. a) **procedure** *par de zeros*($b_1b_2 \ldots b_n$: seqüência de bits, $n \geq 2$)
x := b_1
y := b_2
k := 2
while ($k < n$ e ($x \neq 0$ ou $y \neq 0$))
begin
k := *k* + 1
x := *y*
y := b_k
end
if ($x = 0$ e $y = 0$) **then** imprima "SIM"
else imprima "NÃO"

b) $O(n)$ comparações

5. a) e **b)**
procedure *menor e maior* ($a_1, a_2, \ldots, a_n$: inteiros)
min := a_1
max := a_1
for *i* := 2 **to** *n*
begin
if $a_i < min$ **then** *min* := a_i
if $a_i > max$ **then** *max* := a_i
end {*min* é o menor inteiro entre a entrada, e *max* é o maior}

c) $2n - 2$

7. Antes de serem feitas quaisquer comparações, existe uma possibilidade de que cada elemento possa ser o máximo e uma possibilidade de que possa ser o mínimo. Isto significa que existem $2n$ possibilidades diferentes, e $2n - 2$ delas devem ser eliminadas por comparações de elementos, pois precisamos encontrar o único máximo e o único mínimo. Classificamos as comparações de dois elementos como "virgem" ou "não virgem", dependendo do fato de ocorrer ou não de ambos os elementos comparados já terem estado em alguma comparação anterior. Uma comparação virgem elimina a possibilidade de que o maior seja o mínimo e de que o menor seja o máximo; assim, cada comparação virgem elimina duas possibilidades, mas claramente não pode fazer mais. Uma comparação não virgem deve ser entre dois elementos que ainda podem ser o máximo ou dois elementos que ainda podem ser o mínimo, e pelo menos um destes elementos *não* pode ser candidato à outra categoria. Por exemplo, poderíamos comparar x e

y, em que tudo o que sabemos é que x foi eliminado como o mínimo. Se encontrarmos que $x > y$ neste caso, então apenas uma possibilidade terá sido eliminada — agora sabemos que y não é o máximo. Assim, no pior caso, uma comparação não virgem elimina apenas uma possibilidade. (Os casos das outras comparações não virgens são parecidos.) Agora, existem no máximo $\lfloor n/2 \rfloor$ comparações de elementos que não tenham sido comparados antes, cada um removendo duas possibilidades; eles removem 2 $\lfloor n/2 \rfloor$ possibilidades no total. Portanto, precisamos $2n - 2 - 2\ \lfloor n/2 \rfloor$ outras comparações, e, como argumentamos, podemos remover apenas uma possibilidade em cada para encontrar as respostas no pior caso, pois devem ser eliminadas $2n - 2$ possibilidades. Isto nos dá um total de $2n - 2 - 2\ \lfloor n/2 \rfloor + \lfloor n/2 \rfloor$ comparações no total. Mas $2n - 2 - 2\lfloor n/2 \rfloor + \lfloor n/2 \rfloor = 2n - 2 - \lfloor n/2 \rfloor = 2n - 2 + \lceil -n/2 \rceil = \lceil 2n - n/2 \rceil - 2 = \lceil 3n/2 \rceil - 2$, como desejado. **9.** No final da primeira passagem: 3, 1, 4, 5, 2, 6; no final da segunda passagem: 1, 3, 2, 4, 5, 6; no final da terceira passagem: 1, 2, 3, 4, 5, 6; a quarta passagem não encontra nada para trocar e o algoritmo acaba. **11.** Pode haver no máximo n passagens pela lista e cada passagem usa $O(n)$ comparações. Logo, existem $O(n^2)$ comparações no total. **13.** Como $\log n < n$, temos $(n \log n + n^2)^3 \leq (n^2 + n^2)^3 \leq (2n^2)^3 = 8n^6$ para todo $n > 0$. Isto demonstra que $(n \log n + n^2)^3$ é $O(n^6)$, com testemunhas $C = 8$ e $k = 0$. **15.** $O(x^2 2^x)$ **17.** Observe que $\frac{n!}{2^n} = \frac{n}{2} \cdot \frac{n-1}{2} \cdots \frac{3}{2} \cdot \frac{2}{2} \cdot \frac{1}{2} > \frac{n}{2} \cdot 1 \cdot 1 \cdots 1 \cdot \frac{1}{2} = \frac{n}{4}$. **19.** 5, 22, −12, −29 **21.** Como $ac \equiv bc \pmod{m}$, existe um inteiro k tal que $ac = bc + km$. Logo, $a - b = km/c$. Como $a - b$ é um inteiro, $c \mid km$. Considerando $d = \text{mdc}(m, c)$, escreva $c = de$. Como nenhum fator de e divide m/d, segue que $d \mid m$ e $e \mid k$. Assim, $a - b = (k/e)(m/d)$, em que $k/e \in \mathbf{Z}$ e $m/d \in \mathbf{Z}$. Portanto, $a \equiv b \pmod{m/d}$. **23.** 1 **25.** 1 **27.** $(a_n a_{n-1} \ldots a_1 a_0)_{10} = \sum_{k=0}^{n} 10^k a_k \equiv \sum_{k=0}^{n} a_k \pmod 9$, pois $10^k \equiv 1 \pmod 9$ para todo inteiro não negativo k. **29.** Se não, então suponha que $q_1, q_2, \ldots, q_n$ sejam todos os primos da forma $6k + 5$. Seja $Q = 6q_1q_2 \cdots q_n - 1$. Observe que Q é da forma $6k + 5$, em que $k = q_1q_2 \cdots q_n - 1$. Seja $Q = p_1 p_2 \cdots p_i$ a fatoração em primos de Q. Nenhum p_i é 2, 3, ou qualquer q_j, pois o resto quando Q é dividido por 2 é 1, por 3 é 2 e por q_j é $q_j - 1$. Todos os primos ímpares diferentes de 3 são da forma $6k + 1$ ou $6k + 5$, e o produto de primos da forma $6k + 1$ também é desta forma. Portanto, pelo menos um dos p_i deve ser da forma $6k + 5$, uma contradição. **31.** **a)** Não são primos relativos mutuamente. **b)** São primos relativos mutuamente. **c)** São primos relativos mutuamente. **d)** São primos relativos mutuamente. **33.** **a)** A função de decodificação é $g(q) = \overline{a}(q - b)$ **mod** 26, em que $\overline{a}$ é um inverso de a módulo 26. **b)** PLEASE SEND MONEY. **35.** $x \equiv 28 \pmod{30}$ **37.** Lembre que um polinômio não constante pode assumir o mesmo valor apenas um número finito de vezes. Assim, f pode assumir os valores 0 e ± 1 apenas um número finito de vezes, de modo que, se não existir algum y tal que $f(y)$ seja composto, então deve existir algum x_0 tal que $\pm f(x_0)$ é um primo, digamos p. Observe $f(x_0 + kp)$. Quando substituímos $x_0 + kp$ por x no polinômio e o cancelamos, todo termo conterá p, exceto os termos da forma $f(x_0)$. Portanto, $f(x_0 + kp) = f(x_0) + mp = (m \pm 1)p$ para algum inteiro m. Conforme k varia, este valor pode ser 0, p ou $-p$ apenas um número finito de vezes; portanto, deve haver um número composto para algum valor de k. **39.** Suponha que todo inteiro par maior que 2 seja a soma de dois primos, e seja n um inteiro maior que 5. Se n for ímpar, escreva $n = 3 + (n - 3)$ e decomponha $n - 3 = p + q$ na soma de dois primos; se n for par, então escreva $n = 2 + (n - 2)$ e decomponha $n - 2 = p + q$ na soma de dois primos. Para a recíproca, suponha que todo inteiro maior que 5 seja a soma de três primos, e seja n um inteiro par maior que 2. Escreva $n + 2$ como a soma de três primos, um dos quais é necessariamente 2, de modo que $n + 2 = 2 + p + q$, e portanto $n = p + q$.

41. $\mathbf{A}^{4n} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{A}^{4n+1} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $\mathbf{A}^{4n+2} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$, $\mathbf{A}^{4n+3} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, para $n \geq 0$ **43.** Suponha que $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. Seja $\mathbf{B} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$. Como $\mathbf{AB} = \mathbf{BA}$, segue que $c = 0$ e $a = d$. Seja $\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$. Como $\mathbf{AB} = \mathbf{BA}$, segue que $b = 0$. Logo, $\mathbf{A} = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix} = a\mathbf{I}$.

45. **procedure** *multiplicacao de matriz triangular*(**A, B**: matrizes triangulares superiores $n \times n$, $\mathbf{A} = [a_{ij}]$, $\mathbf{B} = [b_{ij}]$)
  for $i := 1$ **to** n
  for $j := i$ **to** n
  begin
    $c_{ij} := 0$
  for $k := i$ **to** j
    $c_{ij} := c_{ij} + a_{ik}b_{kj}$
  end

47. $(\mathbf{AB})(\mathbf{B}^{-1}\mathbf{A}^{-1}) = \mathbf{A}(\mathbf{BB}^{-1})\mathbf{A}^{-1} = \mathbf{AIA}^{-1} = \mathbf{AA}^{-1} = \mathbf{I}$. Analogamente, $(\mathbf{B}^{-1}\mathbf{A}^{-1})(\mathbf{AB}) = \mathbf{I}$. Logo, $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$. **49.** **a)** Seja $\mathbf{A} \odot \mathbf{0} = [b_{ij}]$. Então, $b_{ij} = (a_{i1} \wedge 0) \vee \cdots \vee (a_{ip} \wedge 0) = 0$. Logo, $\mathbf{A} \odot \mathbf{0} = \mathbf{0}$. Analogamente, $\mathbf{0} \odot \mathbf{A} = 0$. **b)** $\mathbf{A} \vee \mathbf{0} = [a_{ij} \vee 0] = [a_{ij}] = \mathbf{A}$. Logo, $\mathbf{A} \vee \mathbf{0} = \mathbf{A}$. Analogamente, $\mathbf{0} \vee \mathbf{A} = \mathbf{A}$. **c)** $\mathbf{A} \wedge \mathbf{0} = [a_{ij} \wedge 0] = [0] = \mathbf{0}$. Logo, $\mathbf{A} \wedge \mathbf{0} = \mathbf{0}$. Analogamente, $\mathbf{0} \wedge \mathbf{A} = \mathbf{0}$. **51.** Supomos que alguém tenha escolhido um inteiro positivo menor que 2^n, que devemos descobrir. Pedimos à pessoa para escrever o número em binário, usando 0s dominantes, se necessário, para fazer com que tenha n bits de comprimento. Perguntamos então "O primeiro bit é um 1?", "O segundo bit é um 1?", "O terceiro bit é um 1?", e assim por diante. Depois de sabermos as respostas para estas n perguntas, saberemos o número, porque saberemos sua expansão binária.

CAPÍTULO 4

Seção 4.1

1. Seja $P(n)$ a afirmação de que o trem pára na estação n. *Passo base:* Nos foi dito que $P(1)$ é verdadeira. *Passo de indução:* Nos foi dito que $P(n)$ implica $P(n+1)$ para cada $n \geq 1$. Portanto, pelo princípio da indução matemática, $P(n)$ é verdadeira para todos os inteiros positivos n. **3. a)** $1^2 = 1 \cdot 2 \cdot 3/6$ **b)** Ambos os lados de $P(1)$ mostrados na parte (a) são iguais a 1. **c)** $1^2 + 2^2 + \cdots + k^2 = k(k+1)(2k+1)/6$ **d)** Para cada $k \geq 1$, que $P(k)$ implica $P(k+1)$; em outras palavras, que, supondo a hipótese de indução [veja a parte (c)], podemos mostrar que $1^2 + 2^2 + \cdots + k^2 + (k+1)^2 = (k+1)(k+2)(2k+3)/6$ **e)** $(1^2 + 2^2 + \cdots + k^2) + (k+1)^2 = [k(k+1)(2k+1)/6] + (k+1)^2 = [(k+1)/6][k(2k+1) + 6(k+1)] = [(k+1)/6](2k^2 + 7k + 6) = [(k+1)/6](k+2)(2k+3) = (k+1)(k+2)(2k+3)/6$ **f)** Completamos ambos, o passo base e o passo de indução, de modo que, pelo princípio da indução matemática, a afirmação é verdadeira para todo inteiro positivo n. **5.** Seja $P(n)$ a afirmação "$1^2 + 3^2 + \cdots + (2n+1)^2 = (n+1)(2n+1)(2n+3)/3$." *Passo base:* $P(0)$ é verdadeira porque $1^2 = 1 = (0+1)(2 \cdot 0 + 1)(2 \cdot 0 + 3)/3$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $1^2 + 3^2 + \cdots + (2k+1)^2 + [2(k+1)+1]^2 = (k+1)(2k+1)(2k+3)/3 + (2k+3)^2 = (2k+3)[(k+1)(2k+1)/3 + (2k+3)] = (2k+3)(2k^2 + 9k + 10)/3 = (2k+3)(2k+5)(k+2)/3 = [(k+1)+1][2(k+1)+1][2(k+1)+3]/3$. **7.** Seja $P(n)$ a afirmação "$\sum_{j=0}^{n} 3 \cdot 5^j = 3(5^{n+1} - 1)/4$". *Passo base:* $P(0)$ é verdadeira porque $\sum_{j=0}^{0} 3 \cdot 5^j = 3 = 3(5^1 - 1)/4$. *Passo de indução:* Suponha que $\sum_{j=0}^{k} 3 \cdot 5^j = 3(5^{k+1} - 1)/4$. Então $\sum_{j=0}^{k+1} 3 \cdot 5^j = (\sum_{j=0}^{k} 3 \cdot 5^j) + 3 \cdot 5^{k+1} = 3(5^{k+1} - 1)/4 + 3 \cdot 5^{k+1} = 3(5^{k+1} + 4 \cdot 5^{k+1} - 1)/4 = 3(5^{k+2} - 1)/4$. **9. a)** $2 + 4 + 6 + \cdots + 2n = n(n+1)$ **b)** *Passo base:* $2 = 1 \cdot (1+1)$ é verdadeira. *Passo de indução:* Suponha que $2 + 4 + 6 + \cdots + 2k = k(k+1)$. Então $(2 + 4 + 6 + \cdots + 2k) + 2(k+1) = k(k+1) + 2(k+1) = (k+1)(k+2)$. **11. a)** $\sum_{j=1}^{n} 1/2^j = (2^n - 1)/2^n$ **b)** *Passo base:* $P(1)$ é verdadeira porque $\frac{1}{2} = (2^1 - 1)/2^1$. *Passo de indução:* Suponha que $\sum_{j=1}^{k} 1/2^j = (2^k - 1)/2^k$. Então $\sum_{j=1}^{k+1} \frac{1}{2^j} = (\sum_{j=1}^{k} \frac{1}{2^j}) + \frac{1}{2^{k+1}} = \frac{2^k - 1}{2^k} + \frac{1}{2^{k+1}} = \frac{2^{k+1} - 2 + 1}{2^{k+1}} = \frac{2^{k+1} - 1}{2^{k+1}}$. **13.** Seja $P(n)$ a afirmação "$1^2 - 2^2 + 3^2 - \cdots + (-1)^{n-1} n^2 = (-1)^{n-1} n(n+1)/2$". *Passo base:* $P(1)$ é verdadeira porque $1^2 = 1 = (-1)^0 1^2$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $1^2 - 2^2 + 3^2 - \cdots + (-1)^{k-1} k^2 + (-1)^k (k+1)^2 = (-1)^{k-1} k(k+1)/2 + (-1)^k (k+1)^2 = (-1)^k (k+1)[-k/2 + (k+1)] = (-1)^k (k+1)[(k/2)+1] = (-1)^k (k+1)(k+2)/2$. **15.** Seja $P(n)$ a afirmação "$1 \cdot 2 + 2 \cdot 3 + \cdots + n(n+1) = n(n+1)(n+2)/3$." *Passo base:* $P(1)$ é verdadeira porque $1 \cdot 2 = 2 = 1(1+1)(1+2)/3$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $1 \cdot 2 + 2 \cdot 3 + \cdots + k(k+1) + (k+1)(k+2) = [k(k+1)(k+2)/3] + (k+1)(k+2) = (k+1)(k+2)[(k/3) + 1] = (k+1)(k+2)(k+3)/3$. **17.** Seja $P(n)$ a afirmação $1^4 + 2^4 + 3^4 + \cdots + n^4 = n(n+1)(2n+1)(3n^2 + 3n - 1)/30$. $P(1)$ é verdadeira porque $1 \cdot 2 \cdot 3 \cdot 5/30 = 1$. Suponha que $P(k)$ seja verdadeira. Então $(1^4 + 2^4 + 3^4 + \cdots + k^4) + (k+1)^4 = k(k+1)(2k+1)(3k^2 + 3k - 1)/30 + (k+1)^4 = [(k+1)/30][k(2k+1)(3k^2 + 3k - 1) + 30(k+1)^3] = [(k+1)/30](6k^4 + 39k^3 + 91k^2 + 89k + 30) = [(k+1)/30](k+2)(2k+3)[3(k+1)^2 + 3(k+1) - 1]$. Isto mostra que $P(k+1)$ é verdadeira. **19. a)** $1 + \frac{1}{4} < 2 - \frac{1}{2}$ **b)** É verdadeira porque $5/4$ é menor que $6/4$. **c)** $1 + \frac{1}{4} + \cdots + \frac{1}{k^2} < 2 - \frac{1}{k}$ **d)** Para cada $k \geq 2$, que $P(k)$ implica $P(k+1)$; em outras palavras, queremos mostrar que, supondo a hipótese de indução [veja a parte (c)], podemos mostrar que $1 + \frac{1}{4} + \cdots + \frac{1}{k^2} + \frac{1}{(k+1)^2} < 2 - \frac{1}{k+1}$ **e)** $1 + \frac{1}{4} + \cdots + \frac{1}{k^2} + \frac{1}{(k+1)^2} < 2 - \frac{1}{k} + \frac{1}{(k+1)^2} = 2 - \left[\frac{1}{k} - \frac{1}{(k+1)^2}\right] = 2 - \left[\frac{k^2 + 2k + 1 - k}{k(k+1)^2}\right] = 2 - \frac{k^2 + k}{k(k+1)^2} - \frac{1}{k(k+1)^2} = 2 - \frac{1}{k+1} - \frac{1}{k(k+1)^2} < 2 - \frac{1}{k+1}$ **f)** Completamos ambos, o passo base e o passo de indução, de modo que, pelo princípio da indução matemática, a afirmação é verdadeira para todo inteiro n maior que 1. **21.** Seja $P(n)$ a afirmação "$2^n > n^2$". *Passo base:* $P(5)$ é verdadeira porque $2^5 = 32 > 25 = 5^2$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira, ou seja, $2^k > k^2$. Então $2^{k+1} = 2 \cdot 2^k > k^2 + k^2 > k^2 + 4k \geq k^2 + 2k + 1 = (k+1)^2$ porque $k > 4$. **23.** Por inspeção, encontramos que a desigualdade $2n + 3 \leq 2^n$ não é válida para $n = 0, 1, 2, 3$. Seja $P(n)$ a proposição de que esta desigualdade vale para o inteiro positivo n. $P(4)$, o caso base, é verdadeira porque $2 \cdot 4 + 3 = 11 \leq 16 = 2^4$. Para o passo de indução, suponha que $P(k)$ seja verdadeira. Então, pela hipótese de indução, $2(k+1) + 3 = (2k+3) + 2 < 2^k + 2$. Mas como $k \geq 1$, $2^k + 2 \leq 2^k + 2^k = 2^{k+1}$. Isto mostra que $P(k+1)$ é verdadeira. **25.** Seja $P(n)$ a afirmação "$1 + nh \leq (1+h)^n$, $h > -1$". *Passo base:* $P(0)$ é verdadeira porque $1 + 0 \cdot h = 1 \leq 1 = (1+h)^0$. *Passo de indução:* Suponha que $1 + kh \leq (1+h)^k$. Então, como $(1+h) > 0$, $(1+h)^{k+1} = (1+h)(1+h)^k \geq (1+h)(1+kh) = 1 + (k+1)h + kh^2 \geq 1 + (k+1)h$. **27.** Seja $P(n)$ a afirmação "$1/\sqrt{1} + 1/\sqrt{2} + 1/\sqrt{3} + \cdots + 1/\sqrt{n} > 2(\sqrt{n+1} - 1)$". *Passo base:* $P(1)$ é verdadeira porque $1 > 2(\sqrt{2} - 1)$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $1 + 1/\sqrt{2} + \cdots + 1/\sqrt{k} + 1/\sqrt{k+1} > 2(\sqrt{k+1} - 1) + 1/\sqrt{k+1}$. Se mostrarmos que $2(\sqrt{k+1} - 1) + 1/\sqrt{k+1} > 2(\sqrt{k+2} - 1)$, segue que $P(k+1)$ é verdadeira. Esta desigualdade é equivalente a $2(\sqrt{k+2} - \sqrt{k+1}) < 1/\sqrt{k+1}$, que é equivalente a $2(\sqrt{k+2} - \sqrt{k+1})(\sqrt{k+2} + \sqrt{k+1}) < \sqrt{k+1}/\sqrt{k+1} + \sqrt{k+2}/\sqrt{k+1}$. Isto é equivalente a $2 < 1 + \sqrt{k+2}/\sqrt{k+1}$, que é claramente verdadeira. **29.** Seja $P(n)$ a afirmação "$H_{2^n} \leq 1 + n$". *Passo base:* $P(0)$ é verdadeira porque $H_{2^0} = H_1 = 1 \leq 1 + 0$. *Passo de indução:* Suponha que $H_{2^k} \leq 1 + k$. Então $H_{2^{k+1}} = H_{2^k} + \sum_{j=2^k+1}^{2^{k+1}} \frac{1}{j} \leq 1 + k + 2^k (\frac{1}{2^{k+1}}) < 1 + k + 1 = 1 + (k+1)$. **31.** *Passo base:* $1^2 + 1 = 2$ é divisível por 2. *Passo de indução:* Suponha a hipótese de indução, de que $k^2 + k$ é divisível por 2. Então $(k+1)^2 + (k+1) = k^2 + 2k + 1 + k + 1 = (k^2 + k) + 2(k+1)$, a soma de um múltiplo de 2 (pela hipótese de indução) e um múltiplo de 2 (por definição), logo, divisível por 2. **33.** Seja $P(n)$ a afirmação "$n^5 - n$ *é divisível por* 5". *Passo base:* $P(0)$ é verdadeira porque $0^5 - 0 = 0$ é divisível por 5. *Passo de indução:* Suponha que $P(k)$ seja verdadeira, isto é, $k^5 - 5$ é divisível por 5. Então $(k$

$+1)^5-(k+1)=(k^5+5k^4+10k^3+10k^2+5k+1)-(k+1)=(k^5-k)+5(k^4+2k^3+2k^2+k)$ também é divisível por 5, porque ambos os termos nesta soma são divisíveis por 5. **35.** Seja $P(n)$ a proposição de que $(2n-1)^2-1$ é divisível por 8. O caso base $P(1)$ é verdadeiro porque $8 \mid 0$. Agora, suponha que $P(k)$ seja verdadeira. Como $[(2(k+1)-1]^2-1=[(2k-1)^2-1]+8k$, $P(k+1)$ é verdadeira porque ambos os termos no lado direito são divisíveis por 8. Isto mostra que $P(n)$ é verdadeira para todos os inteiros positivos n, de modo que m^2-1é divisível por 8 sempre que m for um inteiro positivo ímpar. **37.** *Passo base:* $11^{1+1}+12^{2\cdot1-1}=121+12=133$. *Passo de indução:* Suponha válida a hipótese de indução, de que $11^{n+1}+12^{2n-1}$é divisível por 133. Então, $11^{(n+1)+1}+12^{2(n+1)-1}=11\cdot11^{n+1}+144\cdot12^{2n-1}=11\cdot11^{n+1}+(11+133)\cdot12^{2n-1}=11(11^{n+1}+12^{n-1})+133\cdot12^{2n-1}$. A expressão entre parênteses é divisível por 133 pela hipótese de indução, e, obviamente, o segundo termo é divisível por 133, de modo que a quantidade toda é divisível por 133, como desejado. **39.** *Passo base:* $A_1 \subseteq B_1$ implica tautologicamente que $\bigcap_{j=1}^{1} A_j \subseteq \bigcap_{j=1}^{1} B_j$. *Passo de indução:* Suponha válida a hipótese de indução de que, se $A_j \subseteq B_j$ for $j=1, 2, ..., k$, então $\bigcap_{j=1}^{k} A_j \subseteq \bigcap_{j=1}^{k} B_j$. Queremos mostrar que, se $A_j \subseteq B_j$ para $j=1, 2, ..., k+1$, então $\bigcap_{j=1}^{k+1} A_j \subseteq \bigcap_{j=1}^{k+1} B_j$. Seja x um elemento arbitrário de $\bigcap_{j=1}^{k+1} A_j = \left(\bigcap_{j=1}^{k} A_j\right) \cap A_{k+1}$. Como $x \in \bigcap_{j=1}^{k} A_j$, sabemos pela hipótese de indução que $x \in \bigcap_{j=1}^{k} B_j$; como $x \in A_{k+1}$, sabemos do fato dado que $A_{k+1} \subseteq B_{k+1}$ que $x \in B_{k+1}$. Portanto, $x \in \left(\bigcap_{j=1}^{k} B_j\right) \cap B_{k+1} = \bigcap_{j=1}^{k+1} B_j$. **41.** Seja $P(n)$ a afirmação "$(A_1 \cup A_2 \cup \cdots \cup A_n) \cap B = (A_1 \cap B) \cup (A_2 \cap B) \cup \cdots \cup (A_n \cap B)$". *Passo base:* $P(1)$ é trivialmente verdadeira. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então, $(A_1 \cup A_2 \cup \cdots \cup A_k \cup A_{k+1}) \cap B = [(A_1 \cup A_2 \cup \cdots \cup A_k) \cup A_{k+1}] \cap B = [(A_1 \cup A_2 \cup \cdots \cup A_k) \cap B] \cup (A_{k+1} \cap B) = [(A_1 \cap B) \cup (A_2 \cap B) \cup \cdots \cup (A_k \cap B)] \cup (A_{k+1} \cap B) = (A_1 \cap B) \cup (A_2 \cap B) \cup \cdots \cup (A_k \cap B) \cup (A_{k+1} \cap B)$. **43.** Seja $P(n)$ a afirmação "$\overline{\bigcup_{k=1}^{n} A_k} = \bigcap_{k=1}^{n} \overline{A_k}$". *Passo base:* $P(1)$ é trivialmente verdadeira. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então, $\overline{\bigcup_{j=1}^{k+1} A_j} = \overline{\left(\bigcup_{j=1}^{k} A_j\right) \cup A_{k+1}} = \overline{\left(\bigcup_{j=1}^{k} A_j\right)} \cap \overline{A_{k+1}} = \left(\bigcap_{j=1}^{k} \overline{A_j}\right) \cap \overline{A_{k+1}} = \bigcap_{j=1}^{k+1} \overline{A_j}$. **45.** Seja $P(n)$ a afirmação de que um conjunto com n elementos tem $n(n-1)/2$ subconjuntos de dois elementos. $P(2)$, o caso base, é verdadeira, porque um conjunto com dois elementos tem um subconjunto com dois elementos — ou seja, ele próprio — e $2(2-1)/2=1$. Agora, suponha que $P(k)$ seja verdadeira. Seja S um conjunto com $k+1$ elementos. Escolha um elemento a em S e seja $T=S-\{a\}$. Um subconjunto de dois elementos de S ou contém a ou não. Aqueles subconjuntos que não contêm a são os subconjuntos de T com dois elementos; pela hipótese de indução, existem $k(k-1)/2$ destes. Existem k subconjuntos de S com dois elementos que contêm a, porque tal subconjunto contém a e um dos k elementos em T. Logo, existem $k(k-1)/2+k=(k+1)k/2$ subconjuntos de dois elementos de S. Isto completa a demonstração por indução. **47.** Os dois conjuntos não se superpõem se $n+1=2$. De fato, a afirmação condicional $P(1) \rightarrow P(2)$ é falsa. **49.** O erro está em aplicar a hipótese de indução para olhar o $\max(x-1, y-1)$, porque, embora x e y sejam inteiros positivos, $x-1$ e $y-1$ não precisam ser (um ou ambos poderiam ser 0). **51.** Usamos a notação (i, j) para indicar o quadrado na linha i e coluna j e indução em $i+j$ para mostrar que todo quadrado pode ser atingido por um cavalo. *Passo base:* Existem seis casos base, para os casos nos quais $i+j \leq 2$. O cavalo já está em (0, 0) no começo, de modo que a seqüência vazia de movimentos atinge o quadrado. Para atingir (1, 0), o cavalo se move (0, 0) → (2, 1) → (0, 2) → (1, 0). Analogamente, para atingir (0, 1), o cavalo se move (0, 0) → (1, 2) → (2, 0) →(0, 1). Observe que o cavalo já atingiu (2, 0) e (0, 2) no processo. Para o último passo base, existe (0, 0) → (1, 2) → (2, 0) → (0, 1) → (2, 2) → (0, 3) → (1, 1). *Passo de indução:* Suponha válida a hipótese de indução, de que o cavalo pode atingir qualquer quadrado (i, j) para o qual $i+j=k$, em que k é um inteiro maior que 1. Devemos mostrar que o cavalo pode atingir cada quadrado (i, j) quando $i+j=k+1$. Como $k+1 \geq 3$, pelo menos um entre i e j é pelo menos 2. Se $i \geq 2$, então, pela hipótese de indução, existe uma seqüência de movimentos que termina em $(i-2, j+1)$, porque $i-2+j+1=i+j-1=k$; de lá, é apenas um passo até (i, j); analogamente, se $j \geq 2$. **53.** *Passo base:* Os casos base $n=0$ e $n=1$ são verdadeiros porque a derivada de x^0 é 0 e a derivada de $x^1=x$ é 1. *Passo de indução:* Usar a regra do produto, a hipótese de indução e o passo base mostra que $\frac{d}{dx}x^{k+1}=\frac{d}{dx}(x\cdot x^k)=x\cdot\frac{d}{dx}x^k+x^k\frac{d}{dx}x=x\cdot kx^{k-1}+x^k\cdot1=kx^k+x^k=(k+1)x^k$. **55.** *Passo base:* Para $k=0, 1 \equiv 1 \pmod m$. *Passo de indução:* Suponha que $a \equiv b \pmod m$ e $a^k \equiv b^k \pmod m$; devemos mostrar que $a^{k+1} \equiv b^{k+1} \pmod m$. Pelo Teorema 5 da Seção 3.4, $a \cdot a^k \equiv b \cdot b^k \pmod m$, o que, por definição, diz que $a^{k+1} \equiv b^{k+1} \pmod m$. **57.** Seja $P(n)$ a afirmação "$[(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_3) \wedge \cdots \wedge (p_{n-1} \rightarrow p_n)] \rightarrow [(p_1 \wedge \cdots \wedge p_{n-1}) \rightarrow p_n]$". *Passo base:* $P(2)$ é verdadeira porque $(p_1 \rightarrow p_2) \rightarrow (p_1 \rightarrow p_2)$ é uma tautologia. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Para mostrar que $[(p_1 \rightarrow p_2) \wedge \cdots \wedge (p_{k-1} \rightarrow p_k) \wedge (p_k \rightarrow p_{k+1})] \rightarrow [(p_1 \wedge \cdots \wedge p_{k-1} \wedge p_k) \rightarrow p_{k+1}]$ é uma tautologia, suponha que a hipótese desta afirmação condicional seja verdadeira. Como ambas as hipóteses e $P(k)$ são verdadeiras, segue que $(p_1 \wedge \cdots \wedge p_{k-1}) \rightarrow p_k$ é verdadeira. Como isto é verdade, e como $p_k \rightarrow p_{k+1}$ é verdadeira (é parte da hipótese), segue por silogismo hipotético que $(p_1 \wedge \cdots \wedge p_{k-1}) \rightarrow p_{k+1}$ é verdadeira. A afirmação mais fraca $(p_1 \wedge \cdots \wedge p_{k-1} \wedge p_k) \rightarrow p_{k+1}$ segue disto. **59.** Demonstraremos primeiro o resultado quando n for uma potência de 2, isto é, se $n=2^k$, $k=1, 2, \ldots$. Seja $P(k)$ a afirmação $A \geq G$, em que A e G são as médias aritmética e geométrica, respectivamente, de um conjunto de $n=2^k$ números reais positivos. *Passo base:* $k=1$ e $n=2^1=2$. Observe que $(\sqrt{a_1}-\sqrt{a_2})^2 \geq 0$. A expansão disto mostra que $a_1-2\sqrt{a_1a_2}+a_2 \geq 0$, ou seja, $(a_1+a_2)/2 \geq (a_1a_2)^{1/2}$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira, com $n=2^k$. Mostraremos que $P(k+1)$ é verdadeira. Temos $2^{k+1}=2n$. Agora, $(a_1+a_2+\cdots+a_{2n})/(2n)=[(a_1+a_2+\cdots+a_n)/n+(a_{n+1}+a_{n+2}+\cdots+a_{2n})/n]/2$ e, analogamente, $(a_1a_2\cdots a_{2n})^{1/(2n)}=[(a_1\cdots a_n)^{1/n}(a_{n+1}\cdots a_{2n})^{1/n}]^{1/2}$. Para simplificar a notação, indique por $A(x, y, \ldots)$ e $G(x, y, ...)$ a média aritmética e geométrica de $x, y, \ldots$, respectivamente. Além disso, se

$x \le x', y \le y'$, e assim por diante, então $A(x, y, \ldots) \le A(x', y', \ldots)$ e $G(x, y, \ldots) \le G(x', y', \ldots)$. Logo, $A(a_1, \ldots, a_{2n}) = A(A(a_1, \ldots, a_n), A(a_{n+1}, \ldots, a_{2n})) \ge A(G(a_1, \ldots, a_n), G(a_{n+1}, \ldots, a_{2n})) \ge G(G(a_1, \ldots, a_n), G(a_{n+1}, \ldots, a_{2n})) = G(a_1, \ldots, a_{2n})$. Isto termina a demonstração para potências de 2. Agora, se n não for uma potência de 2, seja m a próxima potência de 2 mais alta, e sejam $a_{n+1}, \ldots, a_m$ todos iguais a $A(a_1, \ldots, a_n) = \overline{a}$. Então temos $[(a_1 a_2 \cdots a_n)\overline{a}^{m-n}]^{1/m} \le A(a_1, \ldots, a_m)$, pois m é uma potência de 2. Como $A(a_1, \ldots, a_m) = \overline{a}$, segue que $(a_1 \cdots a_n)^{1/m}\overline{a}^{1-n/m} \le \overline{a}^{n/m}$. Elevando ambos os lados à potência (m/n), obtemos $G(a_1, \ldots, a_n) \le A(a_1, \ldots, a_n)$.

61. *Passo base:* Para $n = 1$, o lado esquerdo é simplesmente $\frac{1}{1}$, que é 1. Para $n = 2$, existem três subconjuntos não vazios $\{1\}$, $\{2\}$ e $\{1, 2\}$, de modo que o lado esquerdo é $\frac{1}{1} + \frac{1}{2} + \frac{1}{1\cdot 2} = 2$. *Passo de indução:* Suponha que a afirmação seja verdadeira para k. O conjunto dos primeiros $k + 1$ inteiros positivos tem muitos subconjuntos não vazios, mas eles caem em três categorias: um subconjunto não vazio dos primeiros k inteiros positivos junto com $k + 1$, um subconjunto não vazio dos primeiros k inteiros positivos, ou simplesmente $\{k + 1\}$. Pela hipótese de indução, a soma da primeira categoria é k. Para a segunda categoria, podemos cancelar um fator $1/(k + 1)$ de cada termo da soma e o que resta é simplesmente k pela hipótese de indução, de modo que esta parte da soma é $k/(k + 1)$. Finalmente, a terceira categoria fornece simplesmente $1/(k + 1)$. Logo, a somatória toda é $k + k/(k + 1) + 1/(k + 1) = k + 1$.

63. *Passo base:* Se $A_1 \subseteq A_2$, então A_1 satisfaz a condição de ser subconjunto de cada conjunto na coleção; caso contrário, $A_2 \subseteq A_1$, de modo que A_2 satisfaz a condição. *Passo de indução:* Suponha válida a hipótese de indução, de que a afirmação condicional é verdadeira para conjuntos k, e que nos foram dados $k + 1$ conjuntos que satisfaçam a condição dada. Pela hipótese de indução, deve haver um conjunto A_i para algum $i \le k$ tal que $A_i \subseteq A_j$ para $1 \le j \le k$. Se $A_i \subseteq A_{k+1}$, então teremos acabado. Caso contrário, sabemos que $A_{k+1} \subseteq A_i$, e isto nos diz que A_{k+1} satisfaz a condição de ser um subconjunto de A_j para $1 \le j \le k + 1$.

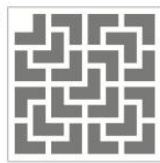
65. $G(1) = 0$, $G(2) = 1$, $G(3) = 3$, $G(4) = 4$.

67. Para mostrar que $2n - 4$ chamadas são suficientes para colocar as fofocas em dia, escolha pessoas 1, 2, 3 e 4 para ser o comitê central. Toda pessoa fora do comitê central chama uma pessoa do comitê central. Neste ponto, os membros do comitê central *como um grupo* sabem todos os escândalos. Eles então trocam informações entre eles fazendo as chamadas 1-2, 3-4, 1-3 e 2-4, nesta ordem. Neste ponto, *todo* membro do comitê central sabe todos os escândalos. Finalmente, todo mundo fora do comitê central chama uma pessoa do comitê central, em cujo ponto todas as pessoas sabem todos os escândalos. [O número total de chamadas é $(n - 4) + 4 + (n - 4) = 2n - 4$.] Que isto não pode ser feito com menos de $2n - 4$ chamadas é muito mais difícil de demonstrar; veja o website http://www.cs.cornell.edu/vogels/Epidemics/gossips-telephones.pdf para detalhes.

69. Demonstramos isto por indução matemática. O passo base ($n = 2$) é verdadeiro tautologicamente. Para $n = 3$, suponha que os intervalos sejam (a, b), (c, d) e (e, f), onde sem perda de generalidade podemos supor que $a \le c \le e$. Como $(a, b) \cap (e, f) \ne \emptyset$, devemos ter $e < b$; por um argumento parecido, $e < d$. Segue que o número no meio do caminho entre e e o menor entre b e d é comum a todos os três intervalos. Agora, para o passo de indução, suponha que, sempre que tivermos k intervalos que têm interseções duas a duas não vazias, então existe um ponto comum a todos os intervalos, e suponha que nos foram dados intervalos $I_1, I_2, \ldots, I_{k+1}$ que têm interseções duas a duas não vazias. Para cada i de 1 a k, seja $J_i = I_i \cap I_{k+1}$. Afirmamos que a coleção $J_1, J_2, \ldots, J_k$ satisfaz a hipótese de indução, ou seja, que $J_{i_1} \cap J_{i_2} \ne \emptyset$ para cada escolha de índices i_1 e i_2. Isto segue do caso $n = 3$ demonstrado acima, usando os conjuntos I_{i_1}, I_{i_2} e I_{k+1}. Podemos agora usar a hipótese de indução para concluir que existe um número comum a todos os conjuntos J_i para $i = 1, 2, \ldots, k$, que obrigatoriamente está na interseção de todos os conjuntos I_i para $i = 1, 2, \ldots, k + 1$.

71. Emparelhe as pessoas. Faça as pessoas ficarem a pequenas distâncias diferentes de seus parceiros, mas bem longe de todas as outras pessoas. Então cada pessoa joga uma torta em seu parceiro, de modo que todo mundo é atingido.

73.



75. Seja $P(n)$ a afirmação de que todo tabuleiro $2^n \times 2^n \times 2^n$ com um cubo $1 \times 1 \times 1$ removido pode ser preenchido pelos tijolos que são cubos $2 \times 2 \times 2$ cada um com um cubo $1 \times 1 \times 1$ removido. O passo base, $P(1)$, é válido porque um tijolo coincide com o sólido a ser recoberto. Agora, suponha que $P(k)$ seja válida. Agora, considere um cubo $2^{k+1} \times 2^{k+1} \times 2^{k+1}$ com um cubo $1 \times 1 \times 1$ removido. Divida este objeto em oito pedaços usando planos paralelos às suas faces e passando pelo centro. O pedaço $1 \times 1 \times 1$ que falta ocorre em uma destas oito partes. Agora, posicione cada tijolo com seu centro no centro do objeto maior, de modo que o cubo $1 \times 1 \times 1$ que falta fique no octante no qual o objeto maior tem um cubo $1 \times 1 \times 1$ faltando. Isto cria oito cubos $2^k \times 2^k \times 2^k$, em cada um faltando um cubo $1 \times 1 \times 1$. Pela hipótese de indução, podemos preencher cada um destes oito objetos com tijolos. Juntando estes recobrimentos, obtemos o preenchimento desejado.

77.

79. Seja $Q(n)$ a afirmação $P(n + b - 1)$. A afirmação de que $P(n)$ é verdadeira para $n = b, b + 1, b + 2, \ldots$ é o mesmo que a afirmação de que $Q(m)$ é verdadeira para todos os inteiros positivos m. Nos foi dado que $P(b)$ é verdadeira [isto é, que $Q(1)$ é verdadeira], e que $P(k) \to P(k + 1)$ para todos $k \ge b$

[isto é, que $Q(m) \rightarrow Q(m+1)$ para todos os inteiros positivos m]. Portanto, pelo princípio da indução matemática, $Q(m)$ é verdadeira para todos os inteiros positivos m.

Seção 4.2

1. *Passo base:* Nos foi dito que podemos correr um quilômetro, de modo que $P(1)$ é verdadeira. *Passo de indução:* Suponha válida a hipótese de indução, de que podemos correr qualquer número de quilômetros entre 1 e k. Devemos mostrar que podemos correr $k+1$ quilômetros. Se $k=1$, então já nos foi dito que podemos correr dois quilômetros. Se $k>1$, então a hipótese de indução nos diz que podemos correr $k-1$ quilômetros, de modo que podemos correr $(k-1)+2=k+1$ quilômetros. **3. a)** $P(8)$ é verdadeira, porque podemos formar 8 centavos em selos com um selo de 3 centavos e um selo de 5 centavos. $P(9)$ é verdadeira, porque podemos formar 9 centavos em selos com três selos de 3 centavos. $P(10)$ é verdadeira, porque podemos formar 10 centavos em selos com dois selos de 5 centavos. **b)** A afirmação de que, usando apenas selos de 3 centavos e de 5 centavos, podemos formar j centavos em selos para todos j com $8 \leq j \leq k$, em que supomos que $k \geq 10$. **c)** Supondo válida a hipótese de indução, podemos formar $k+1$ centavos em selos usando apenas selos de 3 centavos e de 5 centavos. **d)** Como $k \geq 10$, sabemos que $P(k-2)$ é verdadeira, ou seja, que podemos formar $k-2$ centavos em selos. Ponha mais um selo de 3 centavos no envelope, e teremos formado $k+1$ centavos em selos. **e)** Completamos tanto o passo base quanto o passo de indução, de modo que, pelo princípio de indução completa, a afirmação é verdadeira para todo inteiro n maior que ou igual a 8. **5.** a) 4, 8, 11, 12, 15, 16, 19, 20, 22, 23, 24, 26, 27, 28 e todos os valores maiores que ou iguais a 30. **b)** Seja $P(n)$ a afirmação de que podemos formar n centavos em selos usando apenas selos de 4 centavos e de 11 centavos. Queremos demonstrar que P(n) é verdadeira para todo $n \geq 30$. Para o passo base, $30 = 11+11+4+4$. Suponha que possamos formar k centavos em selos (a hipótese de indução); mostraremos como formar $k+1$ centavos em selos. Se os k centavos incluírem um selo de 11 centavos, então substitua-o por três selos de 4 centavos. Caso contrário, os k centavos foram formados usando apenas selos de 4 centavos. Como $k \geq 30$, devem existir pelo menos oito selos de 4 centavos envolvidos. Substitua oito selos de 4 centavos por três selos de 11 centavos e teremos formado $k+1$ centavos em selos. **c)** $P(n)$ é o mesmo que na parte (b). Para demonstrar que $P(n)$ é verdadeira para todo $n \geq 30$, verificamos o passo base de que, $30 = 11+11+4+4$, $31 = 11+4+4+4+4+4$, $32 = 4+4+4+4+4+4+4+4$ e $33 = 11+11+11$. Pelo passo de indução, suponha válida a hipótese de indução, de que $P(j)$ seja verdadeira para todo j com $30 \leq j \leq k$, em que k é um inteiro arbitrário maior que ou igual a 33. Queremos mostrar que $P(k+1)$ é verdadeira. Como $k-3 \geq 30$, sabemos que $P(k-3)$ é verdadeira, ou seja, podemos formar $k-3$ centavos em selos. Ponha mais um selo de 4 centavos no envelope e teremos formado $k+1$ centavos em selos. Nesta demonstração, nossa hipótese de indução era de que $P(j)$ era verdadeira para todos os valores de j entre 30 e k, inclusive, em vez de apenas $P(30)$ ser verdadeira. **7.** Podemos formar todas as quantias, exceto \$ 1 e \$ 3. Seja $P(n)$ a afirmação de que podemos formar n dólares usando apenas notas de 2 e de 5 dólares. Queremos demonstrar que $P(n)$ é verdadeira para todo $n \geq 5$. (É claro que \$ 1 e \$ 3 não podem ser formados e que \$ 2 e \$ 4 podem) Para o passo base, observe que $5 = 5$ e que $6 = 2+2+2$. Suponha válida a hipótese de indução, de que $P(j)$ seja verdadeira para todo j com $5 \leq j \leq k$, em que k é um inteiro arbitrário maior que ou igual a 6. Queremos mostrar que $P(k+1)$ é verdadeira. Como $k-1 \geq 5$, sabemos que $P(k-1)$ é verdadeira, ou seja, que podemos formar $k-1$ dólares. Adicione mais uma outra nota de 2 dólares e teremos formado $k+1$ dólares. **9.** Seja $P(n)$ a afirmação de que não existe nenhum inteiro positivo b, tal que $\sqrt{2} = n/b$. *Passo base:* $P(1)$ é verdadeira porque $\sqrt{2} > 1 \geq 1/b$ para todo inteiro positivo b. *Passo de indução:* Suponha que $P(j)$ seja verdadeira para todo $j \leq k$, em que k é um inteiro positivo arbitrário; demonstramos que $P(k+1)$ é verdadeira por contradição. Suponha que $\sqrt{2} = (k+1)/b$ para algum inteiro positivo b. Então $2b^2 = (k+1)^2$, de modo que $(k+1)^2$ é par, e, portanto, $k+1$ é par. Assim, escreva $k+1 = 2t$ para algum inteiro positivo t, de modo que $2b^2 = 4t^2$ e $b^2 = 2t^2$. Pelo mesmo raciocínio que antes, b é par, de modo que $b = 2s$ para algum inteiro positivo s. Então, $\sqrt{2} = (k+1)/b = (2t)/(2s) = t/s$. Mas $t \leq k$, de modo que isto contradiz a hipótese de indução, e nossa demonstração do passo de indução está completa. **11.** *Passo base:* Existem quatro casos base. Se $n = 1 = 4 \cdot 0 + 1$, então, claramente, o segundo jogador ganha. Se existirem duas, três ou quatro cartas ($n = 4 \cdot 0 + 2$, $n = 4 \cdot 0 + 3$ ou $n = 4 \cdot 1$), então o primeiro jogador pode ganhar removendo todas, exceto uma carta. *Passo de indução:* Suponha válida a hipótese de indução completa, de que em jogos com k ou menos cartas, o primeiro jogador pode ganhar se $k \equiv 0$, 2 ou 3 (mod 4) e o segundo jogador pode ganhar se $k \equiv 1$ (mod 4). Suponha que temos um jogo com $k+1$ cartas, com $k \geq 4$. Se $k+1 \equiv 0$ (mod 4), então o primeiro jogador pode remover 3 cartas, deixando $k-2$ cartas para o outro jogador. Como $k-2 \equiv 1$ (mod 4), pela hipótese de indução, este é um jogo em que o segundo jogador, neste ponto (que é o primeiro jogador em nosso jogo), pode ganhar. Analogamente, se $k+1 \equiv 2$ (mod 4), então o primeiro jogador pode remover uma carta; e, se $k+1 \equiv 3$ (mod 4), então o primeiro jogador pode remover duas cartas. Finalmente, se $k+1 \equiv 1$ (mod 4), então o primeiro jogador deve deixar k, $k-1$ ou $k-2$ cartas para o outro jogador. Como $k \equiv 0$ (mod 4), $k-1 \equiv 3$ (mod 4) e $k-2 \equiv 2$ (mod 4), então, pela hipótese de indução, este é um jogo em que o primeiro jogador, naquele ponto (que é o segundo jogador em nosso jogo), pode ganhar. **13.** Seja $P(n)$ a afirmação de que são necessários exatamente $n-1$ movimentos para montar um quebra-cabeças com n peças. Agora, $P(1)$ é trivialmente verdadeira. Suponha que $P(j)$ seja verdadeira para todos $j \leq k$, e considere um quebra-cabeças com $k+1$ peças. O movimento final deve ser a ligação de dois blocos, de tamanho j e $k+1-j$ para algum inteiro j com $1 \leq j \leq k$. Pela hipótese de indução, são necessários $j-1$ movimentos para construir um bloco, e $k+1-j-1 = k-j$ movimentos para construir o outro. Portanto, são necessários $1+(j-1)+(k-j) = k$ movimentos no total, de modo que $P(k+1)$ é verdadeira. **15.** Considere o tabuleiro de Chomp com n linhas

e n colunas. Afirmamos que o primeiro jogador pode ganhar o jogo fazendo o primeiro movimento para deixar apenas a linha de cima e a coluna mais à esquerda. Seja $P(n)$ a afirmação de que, se um jogador apresentar a um oponente a configuração Chomp com apenas n biscoitos na linha de cima e n biscoitos na coluna mais à esquerda, então ele pode ganhar o jogo. Demonstraremos que $\forall n\, P(n)$ por indução completa. Sabemos que $P(1)$ é verdadeira, porque o oponente é forçado a pegar o biscoito envenenado em sua primeira jogada. Fixe $k \geq 1$ e suponha que $P(j)$ seja verdadeira para todo $j \leq k$. Afirmamos que $P(k+1)$ é verdadeira. É a vez de o oponente fazer um movimento. Se ele escolhe o biscoito envenenado, então o jogo acaba e ele perde. Caso contrário, suponha que ele escolha o biscoito na linha de cima na coluna j, ou o biscoito na coluna mais à esquerda na linha j, para algum j com $2 \leq j \leq k+1$. O primeiro jogador agora escolhe o biscoito na coluna esquerda na linha j, ou o biscoito na linha de cima na coluna j, respectivamente. Isto deixa a posição coberta por $P(j-1)$ para seu oponente, de modo que, pela hipótese de indução, ele pode ganhar. **17.** Seja $P(n)$ a afirmação de que, se um polígono simples com n lados for triangulado, então pelo menos dois dos triângulos na triangulação têm dois lados vizinhos ao exterior do polígono. Demonstraremos que $\forall n \geq 4\ P(n)$. A afirmação é claramente verdadeira para $n = 4$, porque existe apenas uma diagonal, deixando dois triângulos com a propriedade desejada. Fixe $k \geq 4$ e suponha que $P(j)$ seja verdadeira para todo j com $4 \leq j \leq k$. Considere um polígono com $k+1$ lados, e alguma triangulação dele. Escolha uma das diagonais nesta triangulação. Primeiro suponha que esta diagonal divida o polígono em um triângulo e um polígono com k lados. Então, o triângulo tem dois lados na fronteira com o exterior. Além disso, o polígono de k lados tem, pela hipótese de indução, dois triângulos que têm dois lados na fronteira com o exterior do polígono de k lados, e apenas um destes triângulos pode deixar de ser um triângulo que tem dois lados na fronteira com o exterior do polígono original. O único outro caso é que a diagonal divida o polígono em dois polígonos com j lados e $k+3-j$ lados para algum j com $4 \leq j \leq k-1$. Pela hipótese de indução, cada um destes polígonos tem dois triângulos que têm dois lados na fronteira com seu exterior, e em cada caso apenas um destes triângulos pode deixar de ser um triângulo que tem dois lados na fronteira com o exterior do polígono original. **19.** Seja $P(n)$ a afirmação de que a área de um polígono simples com n lados e vértices, todos em pontos reticulados, é dada por $I(P) + B(P)/2 - 1$. Demonstraremos $P(n)$ para todo $n \geq 3$. Começamos com um lema de aditividade: Se P for um polígono simples com todos os vértices em pontos do reticulado, dividido em polígonos P_1 e P_2 por uma diagonal, então $I(P) + B(P)/2 - 1 = [I(P_1) + B(P_1)/2 - 1] + [I(P_2) + B(P_2)/2 - 1]$. Para demonstrar isso, suponha que existam k pontos do reticulado na diagonal, sem contar suas extremidades. Então, $I(P) = I(P_1) + I(P_2) + k$ e $B(P) = B(P_1) + B(P_2) - 2k - 2$; e o resultado segue de cálculos simples. O que isto diz em particular é que, se a fórmula de Pick der a área correta para P_1 e P_2, então ela deve dar a fórmula correta para P, cuja área é a soma das áreas de P_1 e P_2; e analogamente, se a fórmula de Pick der a área correta de P e de um dos P_i's, então ela deve dar a fórmula correta para o outro P_j. Demonstraremos a seguir o teorema para retângulos cujos lados sejam paralelos aos eixos coordenados. Tal retângulo tem vértices necessariamente em (a, b), (a, c), (d, b) e (d, c), em que a, b, c e d são inteiros com $b < c$ e $a < d$. Sua área é $(c-b)(d-a)$. Além disso, $B = 2(c-b+d-a)$ e $I = (c-b-1)(d-a-1) = (c-b)(d-a) - (c-b) - (d-a) + 1$. Portanto, $I + B/2 - 1 = (c-b)(d-a) - (c-b) - (d-a) + 1 + (c-b+d-a) - 1 = (c-b)(d-a)$, que é a área desejada. Considere a seguir um triângulo retângulo cujos lados sejam paralelos aos eixos coordenados. Este triângulo é a metade de um retângulo do tipo que acabamos de considerar, para o qual a fórmula de Pick é válida, de modo que, pelo lema da aditividade, ela vale também para o triângulo. (Os valores de B e I são os mesmos para cada um dos dois triângulos, de modo que, se a fórmula de Pick desse uma resposta que fosse ou muito pequena ou muito grande, então ela daria uma resposta correspondentemente errada para o retângulo.) Para o próximo passo, considere um triângulo arbitrário com vértices nos pontos reticulados que não seja do tipo já considerado. Coloque-o dentro de um retângulo tão pequeno quanto possível. Existem diversas maneiras possíveis de isso ocorrer, mas, em qualquer caso (e adicionando mais uma aresta em um caso), o retângulo terá sido dividido no triângulo dado e em dois ou três triângulos retângulos com lados paralelos aos eixos coordenados. Novamente pelo lema da aditividade, temos certeza de que a fórmula de Pick dá a área correta para o triângulo dado. Isto completa a demonstração de $P(3)$, o passo base em nossa demonstração por indução completa. Para o passo de indução, dado um polígono arbitrário, use o Lema 1 no texto para dividi-lo em dois polígonos. Então, pelo lema da aditividade acima e pela hipótese de indução, sabemos que a fórmula de Pick dá a área correta para este polígono. **21. a)** Na figura à esquerda, $\angle abp$ é menor, mas $\overline{bp}$ não é uma diagonal interior. **b)** Na figura à direita, $\overline{bd}$ não é uma diagonal interior. **c)** Na figura à direita, $\overline{bd}$ não é uma diagonal interior. **23. a)** Quando tentamos demonstrar o passo de indução e encontramos um triângulo em cada subpolígono com pelo menos dois lados na fronteira com o exterior, pode ocorrer em cada caso que o triângulo do qual temos certeza de fato faz fronteira com a diagonal (que é parte da fronteira daquele polígono). Isto nos deixa sem a certeza de nenhum triângulo tocar a fronteira do polígono *original*. **b)** Demonstramos a afirmação mais forte $\forall n \geq 4\ T(n)$ no Exercício 17. **25. a)** O passo de indução aqui nos permite concluir que $P(3)$, $P(5)$, ... são todas verdadeiras, mas não podemos concluir nada sobre $P(2)$, $P(4)$, ... **b)** $P(n)$ é verdadeira para todos os inteiros positivos n, usando indução completa. **c)** O passo de indução aqui nos permite concluir que $P(2), P(4), P(8), P(16), \ldots$ são todas verdadeiras, mas não podemos concluir nada sobre $P(n)$ quando n não for uma potência de 2. **d)** Isto é indução matemática; podemos concluir que $P(n)$ é verdadeira para todos os inteiros positivos n. **27.** Suponha, para uma demonstração por contradição, que exista algum inteiro positivo n tal que $P(n)$ não seja verdadeira. Seja m o menor inteiro positivo maior que n para o qual $P(m)$ é verdadeira; sabemos que tal m existe porque $P(m)$ é verdadeira para infinitos valores de m. Mas sabemos que $P(m) \rightarrow P(m-1)$, de modo que $P(m-1)$ também é verda-

deira. Assim, $m-1$ não pode ser maior que n, de modo que $m-1=n$ e $P(n)$ é, de fato, verdadeira. Esta contradição mostra que $P(n)$ é verdadeira para todo n. **29.** O erro está em ir do caso base $n=0$ para o próximo caso, $n=1$; não podemos escrever 1 como a soma de dois números naturais menores. **31.** Suponha que a propriedade da boa ordenação seja válida. Suponha que $P(1)$ seja verdadeira e que a afirmação condicional $[P(1) \wedge P(2) \wedge \cdots \wedge P(n)] \rightarrow P(n+1)$ seja verdadeira para todo inteiro positivo n. Seja S o conjunto dos inteiros positivos n para os quais $P(n)$ é falsa. Mostraremos que $S=\emptyset$. Suponha que $S \neq \emptyset$. Então, pela propriedade da boa ordenação, existe um menor inteiro m em S. Sabemos que m não pode ser 1 porque $P(1)$ é verdadeira. Como $n=m$ é o menor inteiro, tal que $P(n)$ é falsa, $P(1), P(2), \ldots, P(m-1)$ são verdadeiras, e $m-1 \geq 1$. Como $[P(1) \wedge P(2) \wedge \cdots \wedge P(m-1)] \rightarrow P(m)$ é verdadeira, segue que $P(m)$ também deve ser verdadeira, o que é uma contradição. Logo, $S=\emptyset$. **33.** Em cada caso, dê uma demonstração por contradição baseada em um "menor contra-exemplo", ou seja, valores de n e k tal que $P(n, k)$ não seja verdadeira e n e k sejam menores em algum sentido. **a)** Escolha um contra-exemplo com $n+k$ tão pequeno quanto possível. Não podemos ter $n=1$ e $k=1$, porque nos foi dado que $P(1, 1)$ é verdadeira. Portanto, ou $n>1$ ou $k>1$. No primeiro caso, pela nossa escolha de contra-exemplo, sabemos que $P(n-1, k)$ é verdadeira. Mas o passo de indução então força $P(n, k)$ a ser verdadeira, uma contradição. **b)** Escolha um contra-exemplo com n tão pequeno quanto possível. Não podemos ter $n=1$, porque nos foi dado que $P(1, k)$ é verdadeira para todo k. Portanto, $n>1$. Pela nossa escolha de contra-exemplo, sabemos que $P(n-1, k)$ é verdadeira. Mas o passo de indução então força $P(n, k)$ a ser verdadeira, uma contradição. **c)** Escolha um contra-exemplo com k tão pequeno quanto possível. Não podemos ter $k=1$, porque nos foi dito que $P(n, 1)$ é verdadeira para todo n. Portanto, $k>1$. Pela nossa escolha de contra-exemplo, sabemos que $P(n, k-1)$ é verdadeira. Mas o passo de indução então força $P(n, k)$ a ser verdadeira, uma contradição. **35.** Seja $P(n)$ a afirmação de que, se $x_1, x_2, \ldots, x_n$ forem n números reais distintos, então são usadas $n-1$ multiplicações para encontrar o produto destes números, não importando como os parênteses são inseridos neste produto. Demonstraremos que $P(n)$ é verdadeira usando indução completa. O caso base $P(1)$ é verdadeiro porque são necessárias $1-1=0$ multiplicações para encontrar o produto de x_1, um produto com apenas um fator. Suponha que $P(k)$ seja verdadeira para $1 \leq k \leq n$. A última multiplicação usada para encontrar o produto dos $n+1$ números reais distintos $x_1, x_2, \ldots, x_n, x_{n+1}$ é uma multiplicação do produto dos primeiros k destes números para algum k e o produto dos últimos $n+1-k$ deles. Pela hipótese de indução, são usadas $k-1$ multiplicações para encontrar o produto de k destes números, não importando como os parênteses foram inseridos no produto destes números, e são usadas $n-k$ multiplicações para encontrar o produto dos outros $n+1-k$ deles, não importando como os parênteses foram inseridos no produto destes números. Como é necessária mais uma multiplicação para encontrar o produto de todos os $n+1$ números, o número total de multiplicações usadas é igual a $(k-1)+(n-k)+1=n$. Logo, $P(n+1)$ é verdadeira. **37.** Suponha que $a=dq+r=dq'+r'$ com $0 \leq r<d$ e $0 \leq r'<d$. Então $d(q-q')=r'-r$. Segue que d divide $r'-r$. Como $-d<r'-r<d$, temos $r'-r=0$. Logo, $r'=r$. Segue que $q=q'$. **39.** Isto é um paradoxo causado pela auto-referência. A resposta é claramente "não". Existe um número finito de palavras em inglês, logo, apenas um número finito de seqüências de 15 palavras ou menos; portanto, apenas um número finito de números inteiros positivos pode ser descrito, não todos eles. **41.** Suponha que a propriedade da boa ordenação fosse falsa. Seja S um conjunto não vazio de inteiros não negativos que não tivesse um menor elemento. Seja $P(n)$ a afirmação "$i \notin S$ para $i=0, 1, \ldots, n$". $P(0)$ é verdadeira porque se $0 \in S$ então S tem um menor elemento, a saber, 0. Agora, suponha que $P(n)$ seja verdadeira. Logo, $0 \notin S$, $1 \notin S, \ldots, n \notin S$. Claramente, $n+1$ não pode estar em S, pois, se estivesse, ele seria seu menor elemento. Logo, $P(n+1)$ é verdadeira. Assim, pelo princípio da indução matemática, $n \notin S$ para todo inteiro não negativo n. Logo, $S=\emptyset$, uma contradição. **43.** Isto segue imediatamente do Exercício 41 (se considerarmos o princípio da indução matemática como axioma) e desse exercício junto com a discussão a seguir do enunciado formal de indução completa no texto, o qual mostrou que a indução completa implica o princípio de indução matemática (se considerarmos a indução completa como axioma).

Seção 4.3

1. a) $f(1)=3, f(2)=5, f(3)=7, f(4)=9$ **b)** $f(1)=3, f(2)=9, f(3)=27, f(4)=81$ **c)** $f(1)=2, f(2)=4, f(3)=16, f(4)=65\,536$ **d)** $f(1)=3, f(2)=13, f(3)=183, f(4)=33\,673$ **3. a)** $f(2)=-1, f(3)=5, f(4)=2, f(5)=17$ **b)** $f(2)=-4, f(3)=32, f(4)=-4096, f(5)=536\,870\,912$ **c)** $f(2)=8, f(3)=176, f(4)=92\,672, f(5)=25\,764\,174\,848$ **d)** $f(2)=-\frac{1}{2}, f(3)=-4, f(4)=\frac{1}{8}, f(5)=-32$ **5. a)** Não é válida. **b)** $f(n)=1-n$. *Passo base:* $f(0)=1=1-0$. *Passo de indução:* se $f(k)=1-k$, então $f(k+1)=f(k)-1=1-k-1=1-(k+1)$. **c)** $f(n)=4-n$ se $n>0$ e $f(0)=2$. *Passo base:* $f(0)=2$ e $f(1)=3=4-1$. *Passo de indução* (com $k \geq 1$): $f(k+1)=f(k)-1=(4-k)-1=4-(k+1)$. **d)** $f(n)=2^{\lfloor (n+1)/2 \rfloor}$. *Passo base:* $f(0)=1=2^{\lfloor (0+1)/2 \rfloor}$ e $f(1)=2=2^{\lfloor (1+1)/2 \rfloor}$. *Passo de indução* (com $k \geq 1$): $f(k+1)=2f(k-1)=2 \cdot 2^{\lfloor k/2 \rfloor}=2^{\lfloor k/2 \rfloor+1}=2^{\lfloor ((k+1)+1)/2 \rfloor}$. **e)** $f(n)=3^n$. *Passo base:* Trivial. *Passo de indução:* Para n ímpar, $f(n)=3f(n-1)=3 \cdot 3^{n-1}=3^n$; e para $n>1$ par, $f(n)=9f(n-2)=9 \cdot 3^{n-2}=3^n$. **7.** Existem muitas possibilidades corretas de resposta. Daremos algumas relativamente simples. **a)** $a_{n+1}=a_n+6$ para $n \geq 1$ e $a_1=6$ **b)** $a_{n+1}=a_n+2$ para $n \geq 1$ e $a_1=3$ **c)** $a_{n+1}=10a_n$ para $n \geq 1$ e $a_1=10$ **d)** $a_{n+1}=a_n$ para $n \geq 1$ e $a_1=5$ **9.** $F(0)=0, F(n)=F(n-1)+n$ para $n \geq 1$ **11.** $P_m(0)=0, P_m(n+1)=P_m(n)+m$ **13.** Seja $P(n)$ a afirmação "$f_1+f_3+\cdots+f_{2n-1}=f_{2n}$". *Passo base:* $P(1)$ é verdadeira porque $f_1=1=f_2$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $f_1+f_3+\cdots+f_{2k-1}+f_{2k+1}=f_{2k}+f_{2k+1}=f_{2k+2}+f_{2(k+1)}$. **15.** *Passo base:* $f_0f_1+f_1f_2=0 \cdot 1+1 \cdot 1=1^2=f_2^2$. *Passo de indução:* Suponha que $f_0f_1+f_1f_2+\cdots+f_{2k-1}f_{2k}=f_{2k}^2$. Então $f_0f_1+f_1f_2+\cdots+f_{2k-1}f_{2k}+f_{2k}f_{2k+1}+f_{2k+1}f_{2k+2}=f_{2k}^2+f_{2k}f_{2k+1}+f_{2k+1}f_{2k+2}=f_{2k}$

$(f_{2k} + f_{2k+1}) + f_{2k+1}f_{2k+2} = f_{2k}f_{2k+2} + f_{2k+1}f_{2k+2} = (f_{2k} + f_{2k+1})\, f_{2k+2} = f_{2k+2}^2$. **17.** O número de divisões usadas pelo algoritmo de Euclides para determinar o mdc(f_{n+1}, f_n) é 0 para $n = 0$, 1 para $n = 1$ e $n - 1$ para $n \geq 2$. Para demonstrar este resultado para $n \geq 2$ usamos indução matemática. Para $n = 2$, uma divisão mostra que mdc(f_3, f_2) = mdc(2, 1) = mdc(1, 0) = 1. Agora, suponha que $k - 1$ divisões sejam usadas para encontrar mdc(f_{k+1}, f_k). Para encontrar mdc(f_{k+2}, f_{k+1}), primeiro divida f_{k+2} por f_{k+1} para obter $f_{k+2} = 1 \cdot f_{k+1} + f_k$. Depois de uma divisão temos mdc(f_{k+2}, f_{k+1}) = mdc(f_{k+1}, f_k). Pela hipótese de indução, segue que são necessárias exatamente mais $k - 1$ divisões. Isto mostra que são necessárias k divisões para encontrar mdc(f_{k+2}, f_{k+1}), terminando a demonstração por indução. **19.** $|A| = -1$. Logo, $|A^n| = (-1)^n$. Segue que $f_{n+1}f_{n-1} - f_n^2 = (-1)^n$. **21. a)** Demonstração por indução. *Passo base:* Para $n = 1$, $\max(-a_1) = -a_1 = -\min(a_1)$. Para $n = 2$, existem dois casos. Se $a_2 \geq a_1$, então $-a_1 \geq -a_2$, de modo que $\max(-a_1, -a_2) = -a_1 = -\min(a_1, a_2)$. Se $a_2 < a_1$, então $-a_1 < -a_2$, de modo que $\max(-a_1, -a_2) = -a_2 = -\min(a_1, a_2)$. *Passo de indução:* Suponha que seja verdade para k com $k \geq 2$. Então $\max(-a_1, -a_2, ..., -a_k, -a_{k+1}) = \max(\max(-a_1, ..., -a_k), -a_{k+1}) = \max(-\min(a_1, ..., a_k), -a_{k+1}) = -\min(\min(a_1, ..., a_k), a_{k+1}) = -\min(a_1, ..., a_{k+1})$. **b)** Demonstração por indução matemática. *Passo base:* Para $n = 1$, o resultado é a identidade $a_1 + b_1 = a_1 + b_1$. Para $n = 2$, considere primeiro o caso no qual $a_1 + b_1 \geq a_2 + b_2$. Então $\max(a_1 + b_1, a_2 + b_2) = a_1 + b_1$. Observe também que $a_1 \leq \max(a_1, a_2)$ e $b_1 \leq \max(b_1, b_2)$, de modo que $a_1 + b_1 \leq \max(a_1, a_2) + \max(b_1, b_2)$. Portanto, $\max(a_1 + b_1, a_2 + b_2) = a_1 + b_1 \leq \max(a_1, a_2) + \max(b_1, b_2)$. O caso com $a_1 + b_1 < a_2 + b_2$ é análogo. *Passo de indução:* Suponha que o resultado seja verdadeiro para k. Então $\max(a_1 + b_1, a_2 + b_2, ..., a_k + b_k, a_{k+1} + b_{k+1}) = \max(\max(a_1 + b_1, a_2 + b_2, ..., a_k + b_k), a_{k+1} + b_{k+1}) \leq \max(\max(a_1, a_2, ..., a_k) + \max(b_1, b_2, ..., b_k), a_{k+1} + b_{k+1}) \leq \max(\max(a_1, a_2, ..., a_k), a_{k+1}) + \max(\max(b_1, b_2, ..., b_k), b_{k+1}) = \max(a_1, a_2, ..., a_k, a_{k+1}) + \max(b_1, b_2, ..., b_k, b_{k+1})$. **c)** O mesmo que na parte (b), mas substitua toda ocorrência de "max" por "min" e inverta cada desigualdade. **23.** $5 \in S$ e $x + y \in S$ se $x, y \in S$. **25. a)** $0 \in S$ e, se $x \in S$, então $x + 2 \in S$ e $x - 2 \in S$. **b)** $2 \in S$ e, se $x \in S$, então $x + 3 \in S$. **c)** $1 \in S$, $2 \in S$, $3 \in S$, $4 \in S$ e, se $x \in S$, então $x + 5 \in S$. **27. a)** (0, 1), (1, 1), (2, 1); (0, 2), (1, 2), (2, 2), (3, 2), (4, 2); (0, 3), (1, 3), (2, 3), (3, 3), (4, 3), (5, 3), (6, 3); (0, 4), (1, 4), (2, 4), (3, 4), (4, 4), (5, 4), (6, 4), (7, 4), (8, 4) **b)** Seja $P(n)$ a afirmação de que $a \leq 2b$ sempre que $(a, b) \in S$ for obtido por n aplicações do passo recursivo. *Passo base:* $P(0)$ é verdadeira, porque o único elemento de S obtido sem nenhuma aplicação do passo recursivo é (0, 0) e, de fato, $0 \leq 2 \cdot 0$. *Passo de indução:* Suponha que $a \leq 2b$ sempre que $(a, b) \in S$ seja obtido por k ou menos aplicações do passo recursivo, e considere um elemento obtido com $k + 1$ aplicações do passo recursivo. Como a aplicação final do passo recursivo a um elemento (a, b) deve ser aplicada a um elemento obtido com menos aplicações do passo recursivo, sabemos que $a \leq 2b$. Adicione $0 \leq 2$, $1 \leq 2$ e $2 \leq 2$, respectivamente, para obter $a \leq 2(b + 1)$, $a + 1 \leq 2(b + 1)$ e $a + 2 \leq 2(b + 1)$, como desejado. **c)** Isto vale para o passo base, porque $0 \leq 0$. Se isto valesse para (a, b), então também valeria para os elementos obtidos de (a, b) no passo recursivo, porque adicionar $0 \leq 2$, $1 \leq 2$ e $2 \leq 2$, respectivamente, a $a \leq 2b$ fornece $a \leq 2(b + 1)$, $a + 1 \leq 2(b + 1)$ e $a + 2 \leq 2(b + 1)$. **29. a)** Defina S por $(1, 1) \in S$, e, se $(a, b) \in S$, então $(a + 2, b) \in S$, $(a, b + 2) \in S$ e $(a + 1, b + 1) \in S$. Todos os elementos colocados em S satisfazem a condição, porque (1, 1) tem uma soma par de coordenadas, e, se (a, b) tiver uma soma par de coordenadas, então o mesmo ocorre com $(a + 2, b)$, $(a, b + 2)$ e $(a + 1, b + 1)$. Reciprocamente, mostramos por indução na soma das coordenadas que, se $a + b$ for par, então $(a, b) \in S$. Se a soma for 2, então $(a, b) = (1, 1)$, e o passo base põe (a, b) em S. Caso contrário, a soma é pelo menos 4, e pelo menos um entre $(a - 2, b)$, $(a, b - 2)$ e $(a - 1, b - 1)$ deve ter coordenadas inteiras positivas cuja soma é um número par menor que $a + b$ e, portanto, deve estar em S. Então uma aplicação do passo recursivo mostra que $(a, b) \in S$. **b)** Defina S por (1, 1), (1, 2) e (2, 1) estão em S, e, se $(a, b) \in S$, então $(a + 2, b)$ e $(a, b + 2)$ estão em S. Para demonstrar que nossa definição funciona, observamos primeiro que (1, 1), (1, 2) e (2, 1) têm coordenadas ímpares, e, se (a, b) tiver uma coordenada ímpar, então o mesmo ocorre com $(a + 2, b)$ e $(a, b + 2)$. Reciprocamente, mostramos por indução na soma das coordenadas que, se (a, b) tiver pelo menos uma coordenada ímpar, então $(a, b) \in S$. Se $(a, b) = (1, 1)$ ou $(a, b) = (1, 2)$ ou $(a, b) = (2, 1)$, então o passo base põe (a, b) em S. Caso contrário, a ou b é pelo menos 3, de modo que pelo menos um entre $(a - 2, b)$ e $(a, b - 2)$ deve ter coordenadas inteiras positivas cuja soma é menor que $a + b$ e, portanto, deve estar em S. Então, uma aplicação do passo de recursão mostra que $(a, b) \in S$. **c)** $(1, 6) \in S$ e $(2, 3) \in S$, e, se $(a, b) \in S$, então $(a + 2, b) \in S$ e $(a, b + 6) \in S$. Para demonstrar que nossa definição funciona, observamos primeiro que (1, 6) e (2, 3) satisfazem a condição, e, se (a, b) satisfizer a condição, então o mesmo ocorre com $(a + 2, b)$ e $(a, b + 6)$. Reciprocamente, mostramos por indução na soma das coordenadas que, se (a, b) satisfizer a condição, então $(a, b) \in S$. Para somas 5 e 7, os únicos pontos são (1, 6), que o passo base pôs em S, (2, 3), que o passo base pôs em S e (4, 3) = (2 + 2, 3), que está em S por uma aplicação da definição recursiva. Para uma soma maior que 7, ou $a \geq 3$ ou $a \leq 2$ e $b \geq 9$, em cujo caso ou $(a - 2, b)$ ou $(a, b - 6)$ devem ter coordenadas inteiras positivas cuja soma é menor que $a + b$ e satisfaz a condição para estar em S. Então, uma aplicação do passo recursivo mostra que $(a, b) \in S$. **31.** Se x for um conjunto ou uma variável que representa um conjunto, então x é uma fórmula bem formada. Se x e y forem fórmulas bem formadas, então o mesmo ocorre com $\bar{x}$, $(x \cup y)$, $(x \cap y)$ e $(x - y)$. **33. a)** Se $x \in D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, então $m(x) = x$; se $s = tx$, em que $t \in D^*$ e $x \in D$, então $m(s) = \min(m(s), x)$. **b)** Seja $t = wx$, em que $w \in D^*$ e $x \in D$. Se $w = \lambda$, então $m(st) = m(sx) = \min(m(s), x) = \min(m(s), m(x))$ pelo passo recursivo e pelo passo base da definição de m. Caso contrário, $m(st) = m((sw)x) = \min(m(sw), x)$ pela definição de m. Agora, $m(sw) = \min(m(s), m(w))$ pela hipótese de indução da indução estrutural, de modo que $m(st) = \min(\min(m(s), m(w)), x) = \min(m(s), \min(m(w), x))$ pelo significado de min. Mas $\min(m(w), x) = m(wx) = m(t)$ pelo passo recursivo da definição de m. Logo, $m(st) = \min(m(s), m(t))$. **35.** $\lambda^R = \lambda$ e $(ux)^R = xu^R$ para $x \in \Sigma$, $u \in \Sigma^*$. **37.** $w^0 = \lambda$ e $w^{n+1} =$

ww^n. **39.** Quando a seqüência consistir em n 0s seguidos por n 1s para algum inteiro não negativo n. **41.** Seja $P(i)$ a afirmação "$l(w^i) = i \cdot l(w)$". $P(0)$ é verdadeiro porque $l(w^0) = 0 = 0 \cdot l(w)$. Suponha que $P(i)$ seja verdadeiro. Então $l(w^{i+1}) = l(ww^i) = l(w) + l(w^i) = l(w) + i \cdot l(w) = (i + 1) \cdot l(w)$. **43.** *Passo base:* Para a árvore binária completa consistindo apenas na raiz o resultado é verdadeiro porque $n(T) = 1$ e $h(T) = 0$, e $1 \geq 2 \cdot 0 + 1$. *Passo de indução:* Suponha que $n(T_1) \geq 2h(T_1) + 1$ e $n(T_2) \geq 2h(T_2) + 1$. Pelas definições recursivas de $n(T)$ e $h(T)$, temos $n(T) = 1 + n(T_1) + n(T_2)$ e $h(T) = 1 + \max(h(T_1), h(T_2))$. Portanto, $n(T) = 1 + n(T_1) + n(T_2) \geq 1 + 2h(T_1) + 1 + 2h(T_2) + 1 \geq 1 + 2 \cdot \max(h(T_1), h(T_2)) + 2 = 1 + 2(\max(h(T_1), h(T_2)) + 1) = 1 + 2h(T)$. **45.** *Passo base:* $a_{0,0} = 0 = 0 + 0$. *Passo de indução:* Suponha que $a_{m',n'} = m' + n'$ sempre que (m', n') for menor que (m, n) na ordem lexicográfica de $\mathbf{N} \times \mathbf{N}$. Se $n = 0$, então $a_{m,n} = a_{m-1,n} + 1 = m - 1 + n + 1 = m + n$. Se $n > 0$, então $a_{m,n} = a_{m,n-1} + 1 = m + n - 1 + 1 = m + n$. **47. a)** $P_{m,m} = P_m$ porque um número que exceda m não pode ser usado em uma partição de m. **b)** Como existe apenas uma maneira de obter uma partição de 1, a saber, $1 = 1$, segue que $P_{1,n} = 1$. Como existe apenas uma maneira de obter uma partição de m em 1s, $P_{m,1} = 1$. Quando $n > m$, segue que $P_{m,n} = P_{m,m}$ porque um número que exceda m não pode ser usado. $P_{m,m} = 1 + P_{m,m-1}$ porque uma partição extra, a saber, $m = m$, aparece quando m é permitido na partição. $P_{m,n} = P_{m,n-1} + P_{m-n,n}$ se $m > n$, porque uma partição de m em inteiros que não excedam n ou não usa nenhum n, e, portanto, está contada em $P_{m,n-1}$, ou então usa um n e uma partição de $m - n$, e, portanto, está contada em $P_{m-n,n}$. **c)** $P_5 = 7, P_6 = 11$ **49.** Seja $P(n)$ a afirmação "$A(n, 2) = 4$". *Passo base:* $P(1)$ é verdadeira porque $A(1, 2) = A(0, A(1, 1)) = A(0, 2) = 2 \cdot 2 = 4$. *Passo de indução:* Suponha que $P(n)$ seja verdadeira, ou seja, $A(n, 2) = 4$. Então $A(n + 1, 2) = A(n, A(n + 1, 1)) = A(n, 2) = 4$. **51. a)** 16 **b)** 65 536 **53.** Use um argumento de indução dupla para demonstrar a afirmação mais forte: $A(m, k) > A(m, l)$ quando $k > l$. *Passo base:* Quando $m = 0$, a afirmação é verdadeira porque $k > l$ implica que $A(0, k) = 2k > 2l = A(0, l)$. *Passo de indução:* Suponha que $A(m, x) > A(m, y)$ para todos os inteiros não negativos x e y com $x > y$. Mostraremos que isto implica que $A(m + 1, k) > A(m + 1, l)$ se $k > l$. *Passo base:* Quando $l = 0$ e $k > 0$, $A(m + 1, l) = 0$ e ou $A(m + 1, k) = 2$ ou $A(m + 1, k) = A(m, A(m + 1, k - 1))$. Se $m = 0$, isto é $2A(1, k - 1) = 2^k$. Se $m > 0$, isto é maior que 0 pela hipótese de indução. Em todos os casos, $A(m + 1, k) > 0$, e, de fato, $A(m + 1, k) \geq 2$. Se $l = 1$ e $k > 1$, então $A(m + 1, l) = 2$ e $A(m + 1, k) = A(m, A(m + 1, k - 1))$, com $A(m + 1, k - 1) \geq 2$. Logo, pela hipótese de indução, $A(m, A(m + 1, k - 1)) \geq A(m, 2) > A(m, 1) = 2$. *Passo de indução:* Suponha que $A(m + 1, r) > A(m + 1, s)$ para todo $r > s$, $s = 0, 1, ..., l$. Então, se $k + 1 > l + 1$, segue que $A(m + 1, k + 1) = A(m, A(m + 1, k)) > A(m, A(m + 1, k)) = A(m + 1, l + 1)$. **55.** Do Exercício 54 segue que $A(i, j) \geq A(i - 1, j) \geq \cdots \geq A(0, j) = 2j \geq j$. **57.** Seja $P(n)$ a afirmação "$F(n)$ está bem definida". Então $P(0)$ é verdadeira porque $F(0)$ foi especificado. Suponha que $P(k)$ seja verdadeira para todo $k < n$. Então $F(n)$ está bem definida em n porque $F(n)$ é dado em termos de $F(0)$, $F(1)$, ..., $F(n - 1)$. Assim, $P(n)$ é verdadeira para todos os inteiros n. **59. a)** O valor de $F(1)$ é ambíguo. **b)** $F(2)$ não está definido porque $F(0)$ não está definido. **c)** $F(3)$ é ambíguo e $F(4)$ não está definido porque $F(\frac{4}{3})$ não faz sentido. **d)** A definição de $F(1)$ é ambígua porque ambas, a segunda e a terceira cláusula, parecem se aplicar. **e)** $F(2)$ não pode ser calculado porque a tentativa de calcular $F(2)$ fornece $F(2) = 1 + F(F(1)) = 1 + F(2)$. **61. a)** 1 **b)** 2 **c)** 3 **d)** 3 **e)** 4 **f)** 4 **g)** 5 **63.** $f_0^*(n) = \lceil n/a \rceil$ **65.** $f_2^*(n) = \lceil \log \log n \rceil$ para $n \geq 2$, $f_2^*(1) = 0$

Seção 4.4

1. Primeiro, usamos o passo recursivo para escrever $5! = 5 \cdot 4!$. Então, usamos o passo recursivo repetidamente para escrever $4! = 4 \cdot 3!$, $3! = 3 \cdot 2!$, $2! = 2 \cdot 1!$ e $1! = 1 \cdot 0!$. Inserindo o valor de $0! = 1$, e trabalhando de trás para frente nos passos, vemos que $1! = 1 \cdot 1 = 1$, $2! = 2 \cdot 1! = 2 \cdot 1 = 2$, $3! = 3 \cdot 2! = 3 \cdot 2 = 6$, $4! = 4 \cdot 3! = 4 \cdot 6 = 24$ e $5! = 5 \cdot 4! = 5 \cdot 24 = 120$. **3.** Primeiro, como $n = 11$ é ímpar, usamos a cláusula **else** para ver que *mpotencia*(3, 11, 5) = (*mpotencia* (3, 5, 5)^2 **mod** 5 · 3 **mod** 5) **mod** 5. A seguir, usamos a cláusula **else** novamente para ver que *mpotencia* (3, 5, 5) = (*mpotencia* (3, 2, 5)^2 **mod** 5 · 3 **mod** 5) **mod** 5. Então, usamos a cláusula **else if** para ver que *mpotencia* (3, 2, 5) = *mpotencia* (3, 1, 5)^2 **mod** 5. Usando a cláusula **else** novamente, temos *mpotencia* (3, 1, 5) = (*mpotencia* (3, 0, 5)^2 **mod** 5 · 3 **mod** 5) **mod** 5. Finalmente, usando a cláusula **if**, vemos que *mpotencia* (3, 0, 5) = 1. Trabalhando de trás para frente, segue que *mpotencia* (3, 1, 5) = (1^2 **mod** 5 · 3 **mod** 5) **mod** 5 = 3, *mpotencia* (3, 2, 5) = 3^2 **mod** 5 = 4, *mpotencia* (3, 5, 5) = (4^2 **mod** 5 · 3 **mod** 5) **mod** 5 = 3 e, finalmente, *mpotencia* (3, 11, 5) = (3^2 **mod** 5 · 3 **mod** 5) **mod** 5 = 2. Concluímos que 3^{11} **mod** 5 = 2. **5.** Com esta entrada, o algoritmo usa a cláusula **else** para encontrar que mdc(8, 13) = mdc(13 **mod** 8, 8) = mdc(5, 8). Ele usa esta cláusula novamente para determinar que mdc(5, 8) = mdc(8 **mod** 5, 5) = mdc(3, 5), então para obter mdc(3, 5) = mdc(5 **mod** 3, 3) = mdc(2, 3), então mdc(2, 3) = mdc(3 **mod** 2, 2) = mdc(1, 2), e mais uma vez para obter mdc(1, 2) = mdc(2 **mod** 1, 1) = mdc(0, 1). Finalmente, para determinar mdc(0, 1) ele usa o primeiro passo com $a = 0$ para encontrar que mdc(0, 1) = 1. Conseqüentemente, o algoritmo determina que mdc(8, 13) = 1.

7. **procedure** *mult* (n: inteiro positivo, x: inteiro)
  if $n = 1$ **then** *mult* $(n, x) := x$
  else *mult* $(n, x) := x +$ *mult* $(n - 1, x)$

9. **procedure** *soma de impares* (n: inteiro positivo)
  if $n = 1$ **then** *soma de impares* $(n) := 1$
  else *soma de impares*(n) := *soma de impares* $(n - 1) + 2n - 1$

11. **procedure** *menor* (a_1, ..., a_n: inteiros)
  if $n = 1$ **then** *menor* $(a_1, ..., a_n) = a_1$
  else *menor* $(a_1, ..., a_n)$:=
  min(*menor* $(a_1, ..., a_{n-1})$, a_n)

13. **procedure** *fatorialmod*(n, m: inteiros positivos)
  if $n = 1$ **then** *fatorialmod*$(n, m) := 1$
  else *fatorialmod*(n, m) :=
  $(n \cdot$ *fatorialmod*$(n - 1, m))$ **mod** m

15. **procedure** mdc(a, b: inteiros não negativos)
  {$a < b$ é suposto válido}

if $a = 0$ **then** mdc$(a, b) := b$
else if $a = b - a$ **then** mdc$(a, b) := a$
else if $a < b - a$ **then** mdc$(a, b) :=$ mdc$(a, b - a)$
else mdc$(a, b) :=$ mdc$(b - a, a)$

17. **procedure** *multiplica* (x, y: inteiros não negativos)
if $y = 0$ **then** *multiplica* $(x, y) := 0$
else if y é par **then**
multiplica $(x, y) := 2 \cdot$ *multiplica* $(x, y/2)$
else *multiplica* $(x, y) := 2 \cdot$ *multiplica* $(x, (y - 1)/2) + x$

19. Usamos indução completa em a. *Passo base:* se $a = 0$, sabemos que mdc$(0, b) = b$ para todo $b > 0$, e isto é precisamente o que a cláusula **if** faz. *Passo de indução:* Fixe $k > 0$, suponha válida a hipótese de indução — de que o algoritmo funciona corretamente para todos os valores de seu primeiro argumento menor que k — e considere o que acontece com a entrada (k, b), em que $k < b$. Como $k > 0$, a cláusula **else** é executada, e a resposta é o que o algoritmo der como saída para a entrada (b **mod** k, k). Como b **mod** $k < k$, o par de entrada é verdadeiro. Por nossa hipótese de indução, esta saída é de fato mdc(b **mod** k, k), que é igual a mdc(k, b) pelo Lema 1 da Seção 3.6. **21.** Se $n = 1$, então $nx = x$, e o algoritmo retorna x corretamente. Suponha que o algoritmo calcule corretamente kx. Para calcular $(k + 1)x$, ele calcula recursivamente o produto de $k + 1 - 1 = k$ e x, e então soma x. Pela hipótese de indução, ele calcula o produto corretamente, de modo que a resposta retornada é $kx + x = (k + 1)x$, que é correta.

23. **procedure** *quadrado* (n: inteiro não negativo)
if $n = 0$ **then** *quadrado* $(n) := 0$
else *quadrado* $(n) :=$ *quadrado* $(n - 1) + 2(n - 1) + 1$

Seja $P(n)$ a afirmação de que este algoritmo calcula corretamente n^2. Como $0^2 = 0$, o algoritmo funciona corretamente (usando a cláusula **if**) se a entrada for 0. Suponha que o algoritmo funcione corretamente para entrada k. Então, para entrada $k + 1$, ele dá como saída (por causa da cláusula **else**) sua saída quando a entrada é k, mais $2(k + 1 - 1) + 1$. Pela hipótese de indução, sua saída em k é k^2, de modo que sua saída em $k + 1$ é $k^2 + 2(k + 1 - 1) + 1 = k^2 + 2k + 1 = (k + 1)^2$, como desejado. **25.** n multiplicações *versus* 2^n **27.** $O(\log n)$ *versus* n

29. **procedure** $a(n$: inteiro não negativo)
if $n = 0$ **then** $a(n) := 1$
else if $n = 1$ **then** $a(n) := 2$
else $a(n) := a(n - 1) * a(n - 2)$

31. Iterativo

33. **procedure** *iterativo*(n: inteiro não negativo)
if $n = 0$ **then** $z := 1$
else if $n = 1$ **then** $z := 2$
else
begin
$x := 1$
$y := 2$
$z := 3$
for $i := 1$ **to** $n - 2$
begin
$w := x + y + z$
$x := y$
$y := z$
$z := w$
end
end
{z é o n-ésimo termo da seqüência}

35. Primeiro damos um procedimento recursivo e então um procedimento iterativo.

procedure $r(n$: inteiro não negativo)
if $n < 3$ **then** $r(n) := 2n + 1$
else $r(n) = r(n - 1) \cdot (r(n - 2))^2 \cdot (r(n - 3))^3$

procedure $i(n$: inteiro não negativo)
if $n = 0$ **then** $z := 1$
else if $n = 1$ **then** $z := 3$
else
begin
$x := 1$
$y := 3$
$z := 5$
for $i := 1$ **to** $n - 2$
begin
$w := z * y^2 * x^3$
$x := y$
$y := z$
$z := w$
end
end
{z é o n-ésimo termo da seqüência}

A versão iterativa é mais eficiente.

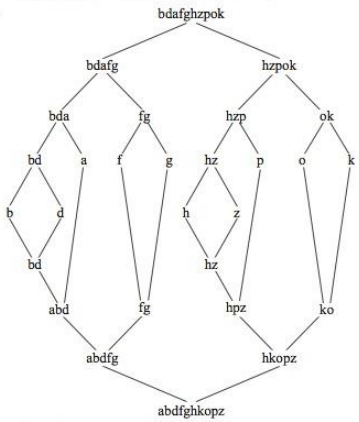
37. **procedure** *inverte* (w: seqüência de bits)
$n :=$ comprimento(w)
if $n \leq 1$ **then** *inverte*$(w) := w$
else *inverte*$(w) :=$
subseq (w, n, n)*inverte* (*subseq* $(w, 1, n - 1)$)
{*subseq* (w, a, b) é a sub-seqüência de w que consiste nos símbolos da posição a até a posição b}

39. O procedimento dá corretamente o inverso de λ como λ (passo base), e, como o inverso de uma seqüência consiste em seu último caractere seguido pelo inverso de seus primeiros $n - 1$ caracteres (veja o Exercício 35 da Seção 4.3), o algoritmo se comporta corretamente quando $n > 0$, pela hipótese de indução. **41.** O algoritmo implementa a idéia do Exemplo 13 da Seção 4.1. Se $n = 1$ (passo base), coloque o único triominó direito de modo que o quadrado entre suas pontas corresponda ao buraco no tabuleiro 2×2. Se $n > 1$, então divida o tabuleiro em quatro tabuleiros, cada um de tamanho $2^{n-1} \times 2^{n-1}$, observe em qual quarto ocorre o buraco, posicione um triominó direito no centro do tabuleiro com o quadrado entre suas pontas no quarto onde estiver o quadrado faltando (veja a Figura 8 na Seção 4.1) e chame o algoritmo recursivamente quatro vezes — uma para cada tabuleiro $2^{n-1} \times 2^{n-1}$, cada um dos quais tem um quadrado faltando (ou porque ele estava faltando no começo, ou porque ele foi coberto pelo triominó central).

43. **procedure** $A(m, n$: inteiros não negativos)
if $m = 0$ **then** $A(m, n) := 2n$
else if $n = 0$ **then** $A(m, n) := 0$
else if $n = 1$ **then** $A(m, n) := 2$

else $A(m, n) := A(m - 1, A(m, n - 1))$

45.



47. Considere as duas listas com $1, 2, \ldots, m - 1, m + n - 1$ e $m, m + 1, \ldots, m + n - 2, m + n$, respectivamente. **49.** Se $n = 1$, então o algoritmo não faz nada, o que está correto porque uma lista com um elemento já está ordenada. Suponha que o algoritmo funcione corretamente para $n = 1$ até $n = k$. Se $n = k + 1$, então a lista é quebrada em duas listas, L_1 e L_2. Pela hipótese de indução, *mergesort* ordena corretamente cada uma destas sublistas; além disso, *merge* funde corretamente as duas listas ordenadas em uma porque com cada comparação o menor elemento em $L_1 \cup L_2$ que ainda não foi posto em L é colocado lá. **51.** $O(n)$ **53.** 6 **55.** $O(n^2)$

Seção 4.5

1. Suponha que $x = 0$. O segmento de programa primeiro atribui o valor 1 a y e então atribui o valor $x + y = 0 + 1 = 1$ a z. **3.** Suponha que $y = 3$. O segmento de programa atribui o valor 2 a x e então atribui o valor $x + y = 2 + 3 = 5$ a z. Como $y = 3 > 0$, ele então atribui o valor $z + 1 = 5 + 1 = 6$ a z.

5.
$$\begin{array}{l} (p \wedge \textit{condição}1)\{S_1\}q \\ (p \wedge \neg \textit{condição}1 \wedge \textit{condição}2)\{S_2\}q \\ \quad . \\ \quad . \\ \quad . \\ (p \wedge \neg \textit{condição}1 \wedge \neg\, \textit{condição}2 \\ \qquad \cdots \wedge \neg \textit{condição}(n-1)\{S_n\}q \\ \hline \therefore\ p\{\textbf{if } \textit{condição}1 \textbf{ then } S_1; \\ \quad \textbf{else if } \textit{condição}2 \textbf{ then } S_2; \ldots; \textbf{else } S_n\}q \end{array}$$

7. Mostraremos que p: "$potencia = x^{i-1}$ e $i \leq n + 1$" é um laço invariável. Observe que p é verdadeira inicialmente, porque antes de o laço começar, $i = 1$ e $potencia = 1 = x^0 = x^{1-1}$. A seguir, devemos mostrar que, se p é verdadeira e $i \leq n$ depois de uma execução do laço, então p permanece verdadeira depois de mais uma execução. O laço aumenta i por 1. Logo, como $i \leq n$ antes desta passagem, $i \leq n + 1$ depois desta passagem. Além disso, o laço atribui $potencia \cdot x$ a $potencia$. Pela hipótese de indução podemos ver que à $potencia$ é atribuído o valor $x^{i-1} \cdot x = x^i$. Logo, p permanece verdadeira. Além disso, o laço termina depois de n passagens do laço com $i = n + 1$ porque a i é atribuído o valor 1 antes de entrar no laço, é aumentado por 1 em cada passagem, e o laço termina quando $i > n$. Conseqüentemente, no final $potencia = x^n$, como desejado. **9.** Suponha que p seja "m e n são inteiros". Então, se a condição $n < 0$ for verdadeira, $a = -n = |n|$ depois de S_1 ser executado. Se a condição $n < 0$ for falsa, então $a = n = |n|$ depois de S_1 ser executado. Logo, $p\{S_1\}q$ é verdadeira onde q é "$a = |n|$". Como S_2 atribui o valor 0 a ambos, k e x, é claro que $q\{S_2\}r$ é verdadeira onde r é $q \wedge (k = 0) \wedge (x = 0)$. Suponha que r seja verdadeira. Seja $P(k)$ a afirmação "$x = mk$ e $k \leq a$". Podemos mostrar que $P(k)$ é um laço invariável para o laço em S_3. $P(0)$ é verdadeira porque antes de entrar no laço $x = 0 = m \cdot 0$ e $0 \leq a$. Agora, suponha que $P(k)$ seja verdadeira e $k < a$. Então $P(k + 1)$ é verdadeira porque a x é atribuído o valor $x + m = mk + m = m(k + 1)$. O laço termina quando $k = a$, e naquele ponto $x = ma$. Logo, $r\{S_3\}s$ é verdadeira, em que s é "$a = |n|$ e $x = ma$". Agora, suponha que s seja verdadeira. Então, se $n < 0$, segue que $a = -n$, de modo que $x = -mn$. Neste caso, S_4 atribui $-x = mn$ a $produto$. Se $n > 0$, então $x = ma = mn$, de modo que S_4 atribui mn a $produto$. Logo, $s\{S_4\}t$ é verdadeira. **11.** Suponha que a afirmação inicial p seja verdadeira. Então, como $p\{S\}q_0$ é verdadeira, q_0 é verdadeira depois de o segmento S ser executado. Como $q_0 \rightarrow q_1$ é verdadeira, segue também que q_1 é verdadeira depois de S ser executado. Logo, $p\{S\}q_1$ é verdadeira. **13.** Usaremos a proposição p, "$\text{mdc}(a, b) = \text{mdc}(x, y)$ e $y \geq 0$", como o laço invariável. Observe que p é verdadeira antes de entrar no laço, porque, naquele ponto, $x = a$, $y = b$ e y é um inteiro positivo, usando a afirmação inicial. Agora, suponha que p seja verdadeira e $y > 0$; então o laço será executado novamente. Dentro do laço, x e y são substituídos por y e x **mod** y, respectivamente. Pelo Lema 1 da Seção 3.6, $\text{mdc}(x, y) = \text{mdc}(y, x$ **mod** $y)$. Portanto, depois da execução do laço, o valor de $\text{mdc}(x, y)$ é o mesmo que era antes. Além disso, como y é o resto, ele é pelo menos 0. Logo, p permanece verdadeira, de modo que ela é um laço invariável. Além do mais, se o laço terminar, então $y = 0$. Neste caso, temos $\text{mdc}(x, y) = x$, a afirmação final. Portanto, o programa, que dá x como sua saída, calculou corretamente $\text{mdc}(a, b)$. Finalmente, podemos demonstrar que o laço termina, porque cada iteração faz com que o valor de y decresça pelo menos 1. Portanto, o laço pode ser iterado no máximo b vezes.

Exercícios Complementares

1. Seja $P(n)$ a afirmação de que esta equação é válida. *Passo base:* $P(1)$ diz que $2/3 = 1 - (1/3^1)$, o que é verdade. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $2/3 + 2/9 + 2/27 + \cdots + 2/3^n + 2/3^{n+1} = 1 - 1/3^n + 2/3^{n+1}$ (pela hipótese de indução), e isto é igual a $1 - 1/3^{n+1}$, como desejado. **3.** Seja $P(n)$ a afirmação "$1 \cdot 1 + 2 \cdot 2 + \cdots + n \cdot 2^{n-1} = (n - 1)2^n + 1$". *Passo base:* $P(1)$ é verdadeira porque $1 \cdot 1 = 1 = (1 - 1)2^1 + 1$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $1 \cdot 1 + 2 \cdot 2 + \cdots + k \cdot 2^{k-1} + (k$

$+ 1) \cdot 2^k = (k-1)2^k + 1 + (k+1)2^k = 2k \cdot 2^k + 1 = [(k+1) - 1]2^{k+1} + 1$. **5.** Seja $P(n)$ a afirmação "$1/(1 \cdot 4) + \cdots + 1/[(3n-2)(3n+1)] = n/(3n+1)$". *Passo base:* $P(1)$ é verdadeira porque $1/(1 \cdot 4) = 1/4$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $1/(1 \cdot 4) + \cdots + 1/[(3k-2)(3k+1)] + 1/[(3k+1)(3k+4)] = k/(3k+1) + 1/[(3k+1)(3k+4)] = [k(3k+4)+1]/[(3k+1)(3k+4)] = [(3k+1)(k+1)]/[(3k+1)(3k+4)] = (k+1)/(3k+4)$. **7.** Seja $P(n)$ a afirmação "$2^n > n^3$". *Passo base:* $P(10)$ é verdadeira porque $1024 > 1000$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $(k+1)^3 = k^3 + 3k^2 + 3k + 1 \leq k^3 + 9k^2 \leq k^3 + k^3 = 2k^3 < 2 \cdot 2^k = 2^{k+1}$. **9.** Seja $P(n)$ a afirmação "$a - b$ é um fator de $a^n - b^n$". *Passo base:* $P(1)$ é trivialmente verdadeira. Suponha que $P(k)$ seja verdadeira. Então $a^{k+1} - b^{k+1} = a^{k+1} - ab^k + ab^k - b^{k+1} = a(a^k - b^k) + b^k(a - b)$. Então, como $a - b$ é um fator de $a^k - b^k$ e $a - b$ é um fator de $a - b$, segue que $a - b$ é um fator de $a^{k+1} - b^{k+1}$. **11.** Seja $P(n)$ a afirmação "$a + (a+d) + \cdots + (a+nd) = (n+1)(2a+nd)/2$". *Passo base:* $P(1)$ é verdadeira porque $a + (a+d) = 2a + d = 2(2a+d)/2$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $a + (a+d) + \cdots + (a+kd) + [a + (k+1)d] = (k+1)(2a+kd)/2 + a + (k+1)d = \frac{1}{2}(2ak + 2a + k^2d + kd + 2a + 2kd + 2d) = \frac{1}{2}(2ak + 4a + k^2d + 3kd + 2d) = \frac{1}{2}(k+2)[2a + (k+1)d]$. **13.** *Passo base:* Isto é verdade para $n = 1$ porque $5/6 = 10/12$. *Passo de indução:* Suponha que esta equação seja válida para $n = k$, e considere $n = k+1$. Então $\sum_{i=1}^{k+1} \frac{i+4}{i(i+1)(i+2)} = \sum_{i=1}^{k} \frac{i+4}{i(i+1)(i+2)} + \frac{k+5}{(k+1)(k+2)(k+3)} = \frac{k(3k+7)}{2(k+1)(k+2)} + \frac{k+5}{(k+1)(k+2)(k+3)}$ (pela hipótese de indução) $= \frac{1}{(k+1)(k+2)} \cdot \left(\frac{k(3k+7)}{2} + \frac{k+5}{k+3}\right) = \frac{1}{2(k+1)(k+2)(k+3)} \cdot [k(3k+7)(k+3) + 2(k+5)] = \frac{1}{2(k+1)(k+2)(k+3)} \cdot (3k^3 + 16k^2 + 23k + 10) = \frac{1}{2(k+1)(k+2)(k+3)} \cdot (3k+10)(k+1)^2 = \frac{1}{2(k+2)(k+3)} \cdot (3k+10)(k+1) = \frac{(k+1)(3(k+1)+7)}{2((k+1)+1)((k+1)+2)}$, como desejado. **15.** *Passo base:* A afirmação é verdadeira para $n = 1$ porque a derivada de $g(x) = xe^x$ é $x \cdot e^x + e^x = (x+1)e^x$ pela regra do produto. *Passo de indução:* Suponha que a afirmação seja verdadeira para $n = k$, isto é, a k-ésima derivada é dada por $g^{(k)} = (x+k)e^x$. Derivando pela regra do produto, obtemos a $(k+1)$-ésima derivada: $g^{(k+1)} = (x+k)e^x + e^x = [x + (k+1)]e^x$, como desejado. **17.** Usaremos agora indução completa para mostrar que f_n é par se $n \equiv 0 \pmod 3$ e é ímpar caso contrário. *Passo base:* isto segue porque $f_0 = 0$ é par e $f_1 = 1$ é ímpar. *Passo de indução:* Suponha que, se $j \leq k$, então f_j é par se $j \equiv 0 \pmod 3$ e é ímpar caso contrário. Agora, suponha que $k+1 \equiv 0 \pmod 3$. Então, $f_{k+1} = f_k + f_{k-1}$ é par porque f_k e f_{k-1} são ambos ímpares. Se $k+1 \equiv 1 \pmod 3$, então $f_{k+1} = f_k + f_{k-1}$ é ímpar porque f_k é par e f_{k-1} é ímpar. Finalmente, se $k+1 \equiv 2 \pmod 3$, então $f_{k+1} = f_k + f_{k-1}$ é ímpar porque f_k é ímpar e e f_{k-1} é par. **19.** Seja $P(n)$ a afirmação $f_k f_n + f_{k+1} f_{n+1} = f_{n+k+1}$ para todo inteiro não negativo k. *Passo base:* Isto consiste em mostrar que $P(0)$ e $P(1)$ são ambos válidos. $P(0)$ é verdadeira porque $f_k f_0 + f_{k+1} f_1 = f_{k+1} \cdot 0 + f_{k+1} \cdot 1 = f_1$. Como $f_k f_1 + f_{k+1} f_2 = f_k + f_{k+1} = f_{k+2}$, segue que $P(1)$ é verdadeira. *Passo de indução:* Agora, suponha que $P(j)$ seja válida. Então, pela hipótese de indução e pela definição recursiva dos números de Fibonacci, segue que $f_{k+1} f_{j+1} + f_{k+2} f_{j+2} = f_k(f_{j-1} + f_j) + f_{k+1}(f_j + f_{j+1}) = (f_k f_{j-1} + f_{k+1} f_j) + (f_k f_j + f_{k+1} f_{j+1}) = f_{j-1+k+1} + f_{j+k+1} = f_{j+k+2}$. Isto mostra que $P(j+1)$ é verdadeira. **21.** Seja $P(n)$ a afirmação $l_0^2 + l_1^2 + \cdots + l_n^2 = l_n l_{n+1} + 2$. *Passo base:* $P(0)$ e $P(1)$ são ambas válidas, porque $l_0^2 = 2^2 = 2 \cdot 1 + 2 = l_0 l_1 + 2$ e $l_0^2 + l_1^2 = 2^2 + 1^2 = 1 \cdot 3 + 2 = l_1 l_3 + 2$. *Passo de indução:* Suponha que $P(k)$ seja válida. Então, pela hipótese de indução $l_0^2 + l_1^2 + \cdots + l_k^2 + l_{k+1}^2 = l_k l_{k+1} + 2 + l_{k+1}^2 = l_{k+1}(l_k + l_{k+1}) + 2 = l_{k+1} l_{k+2} + 2$. Isto mostra que $P(k+1)$ é válida. **23.** Seja $P(n)$ a afirmação de que a identidade é válida para todo n. *Passo base:* $P(1)$ é obviamente verdadeira. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então, $\cos(k+1)x + i\,\mathrm{sen}(k+1)x = \cos(kx + x) + i\,\mathrm{sen}(kx + x) = \cos kx \cos x - \mathrm{sen}\, kx\, \mathrm{sen}\, x + i(\mathrm{sen}\, kx \cos x + \cos kx\, \mathrm{sen}\, x) = \cos x\,(\cos kx + i\,\mathrm{sen}\, kx)(\cos x + i\,\mathrm{sen}\, x) = (\cos x + i\,\mathrm{sen}\, x)^k (\cos x + i\,\mathrm{sen}\, x) = (\cos x + i\,\mathrm{sen}\, x)^{k+1}$. Segue que $P(k+1)$ é verdadeira. **25.** Reescreva o lado direito como $2^{n+1}(n^2 - 2n + 3) - 6$. Para $n = 1$ temos $2 = 4 \cdot 2 - 6$. Suponha que a equação é válida para $n = k$, e considere $n = k+1$. Então $\sum_{j=1}^{k+1} j^2 2^j = \sum_{j=1}^{k} j^2 2^j + (k+1)^2 2^{k+1} = 2^{k+1}(k^2 - 2k + 3) - 6 + (k^2 + 2k + 1)2^{k+1}$ (pela hipótese de indução) $= 2^{k+1}(2k^2 + 4) - 6 = 2^{k+2}(k^2 + 2) - 6 = 2^{k+2}[(k+1)^2 - 2(k+1) + 3] - 6$. **27.** Seja $P(n)$ a afirmação de que esta equação é válida. *Passo base:* Em $P(2)$ ambos os lados se reduzem a $1/3$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Então $\sum_{j=1}^{k+1} 1/(j^2 - 1) = \left(\sum_{j=1}^{k} 1/(j^2 - 1)\right) + 1/[(k+1)^2 - 1] = (k-1)(3k+2)/[4k(k+1)] + 1/[(k+1)^2 - 1]$ pela hipótese de indução. Isto se simplifica para $(k-1)(3k+2)/[4k(k+1)] + 1/(k^2 + 2k) = (3k^3 + 5k^2)/[4k(k+1)(k+2)] = \{[(k+1) - 1][3(k+1) + 2]\}/[4(k+1)(k+2)]$, que é exatamente o que $P(k+1)$ afirma. **29.** Seja $P(n)$ a afirmação de que pelo menos $n+1$ retas são necessárias para cobrir o reticulado de pontos na região triangular dada. *Passo base:* $P(0)$ é verdadeira, porque precisamos pelo menos de uma reta para cobrir o ponto em $(0, 0)$. *Passo de indução:* Suponha válida a hipótese de indução, de que pelo menos $k+1$ retas são necessárias para cobrir o reticulado de pontos com $x \geq 0$, $y \geq 0$ e $x + y \leq k$. Considere o triângulo de pontos do reticulado definido por $x \geq 0$, $y \geq 0$ e $x + y \leq k+1$. Por contradição, suponha que $k+1$ retas pudessem cobrir este conjunto. Então, estas retas deveriam cobrir os $k+2$ pontos na reta $x + y = k+1$. Mas apenas a própria reta $x + y = k+1$ pode cobrir mais que um destes pontos, porque duas retas distintas intersectam em, no máximo, em ponto. Portanto, nenhuma das $k+1$ retas que são necessárias (pela hipótese de indução) para cobrir o conjunto de pontos do reticulado dentro do triângulo, mas não nesta reta, pode cobrir mais de um ponto nesta reta, e isto deixa pelo menos um ponto não coberto. Portanto, nossa hipótese de que $k+1$ retas podiam cobrir um conjunto maior está errada, e nossa demonstração está completa. **31.** Seja $P(n)$ a afirmação $\mathbf{B}^k = \mathbf{M}\mathbf{A}^k\mathbf{M}^{-1}$. *Passo base:* Parte das condições dadas. *Passo de indução:* Suponha válida a hipótese de indução. Então $\mathbf{B}^{k+1} = \mathbf{B}\mathbf{B}^k = \mathbf{M}\mathbf{A}\mathbf{M}^{-1}\mathbf{B}^k = \mathbf{M}\mathbf{A}\mathbf{M}^{-1}\mathbf{M}\mathbf{A}^k\mathbf{M}^{-1}$ (pela hipótese de indução) $= \mathbf{M}\mathbf{A}\mathbf{I}\mathbf{A}^k\mathbf{M}^{-1} = \mathbf{M}\mathbf{A}\mathbf{A}^k\mathbf{M}^{-1} = \mathbf{M}\mathbf{A}^{k+1}\mathbf{M}^{-1}$. **33.** Demonstramos por indução matemática a seguinte afirmação mais forte: Para todo $n \geq 3$, podemos escrever $n!$ como a soma de n de seus divisores distintos positivos, um dos quais é 1. Ou seja, pode-

mos escrever $n! = a_1 + a_2 + \cdots + an$, em que cada a_i é um divisor de $n!$, os divisores estão listados em ordem estritamente decrescente e $a_n = 1$. *Passo base:* $3! = 3 + 2 + 1$. *Passo de indução:* Suponha que possamos escrever $k!$ como uma soma da forma desejada, digamos $k! = a_1 + a_2 + \cdots + a_k$, em que cada a_i é um divisor de $n!$, os divisores estão listados em ordem estritamente decrescente e $a_n = 1$. Considere $(k + 1)!$. Então, temos $(k + 1)! = (k + 1)k! = (k + 1)(a_1 + a_2 + \cdots + a_k) = (k + 1)a_1 + (k + 1)a_2 + \cdots + (k + 1)a_k = (k + 1)a_1 + (k + 1)a_2 + \cdots + k \cdot a_k + a_k$. Como cada a_i era um divisor de $k!$, cada $(k + 1)a_i$ é um divisor de $(k + 1)!$. Além disso, $k \cdot a_k = k$, que é um divisor de $(k + 1)!$, e $a_k = 1$, de modo que o novo último somando é novamente 1. (Observe também que nossa lista de somandos ainda está em ordem estritamente decrescente.) Assim, escrevemos $(k + 1)!$ na forma desejada. **35.** Quando $n = 1$, a afirmação é trivialmente verdadeira. Suponha que a afirmação seja verdadeira para $n = k$, e considere $k + 1$ pessoas em fila, com uma mulher em primeiro e um homem por último. Se a k-ésima pessoa for uma mulher, então temos aquela mulher em frente do homem em último lugar. Se a k-ésima pessoa for um homem, então as primeiras k pessoas na fila satisfazem a condição da hipótese de indução para as primeiras k pessoas na fila, de modo que novamente podemos concluir que existe uma mulher diretamente em frente de um homem em algum lugar na fila. **37.** *Passo base:* Quando $n = 1$ existe um círculo, e podemos colorir o interior de azul e o exterior de vermelho para satisfazer as condições. *Passo de indução:* Suponha válida a hipótese de indução de que, se existirem k círculos, então as regiões podem ser coloridas com duas cores, de modo que nenhum par de regiões com uma fronteira comum tem a mesma cor, e considere uma situação com $k + 1$ círculos. Remova um dos círculos, reduzindo uma figura com k círculos, e use a hipótese de indução para colori-la da maneira prescrita. Então, reponha o círculo removido e mude a cor de cada região dentro deste círculo. A figura resultante satisfaz a condição, porque, se duas regiões tiverem uma fronteira comum, então ou esta região envolve o novo círculo, em cujo caso as regiões em cada lado eram a mesma região e, agora, a parte de dentro é diferente da parte de fora, ou então a fronteira não envolve o novo círculo, em cujo caso as regiões estão coloridas de modo diferente porque elas já estavam coloridas de modo diferente antes de o novo círculo ser reposto. **39.** Se $n = 1$, então a equação fica $1 \cdot 1 = 1 \cdot 2/2$, que é verdadeira. Suponha que a equação seja verdadeira para n e considere-a para $n + 1$. Então $\sum_{j=1}^{n+1}(2j-1)\left(\sum_{k=j}^{n+1}\frac{1}{k}\right) = \sum_{j=1}^{n}(2j-1)\left(\sum_{k=j}^{n+1}\frac{1}{k}\right) + [2(n+1)-1]\cdot\frac{1}{n+1} = \sum_{j=1}^{n}(2j-1)\left(\frac{1}{n+1} + \sum_{k=j}^{n}\frac{1}{k}\right) + \frac{2n+1}{n+1} = \left(\frac{1}{n+1}\sum_{j=1}^{n}(2j-1)\right) + \left(\sum_{j=1}^{n}(2j-1)\sum_{k=j}^{n}\frac{1}{k}\right) + \frac{2n+1}{n+1} = \left(\frac{1}{n+1}\cdot n^2\right) + \frac{n(n+1)}{2} + \frac{2n+1}{n+1}$ (pela hipótese de indução) $= \frac{2n^2+n(n+1)^2+(4n+2)}{2(n+1)} = \frac{2(n+1)^2+n(n+1)^2}{2(n+1)} = \frac{(n+1)(n+2)}{2}$.

41. a) 92 **b)** 91 **c)** 91 **d)** 91 **e)** 91 **f)** 91

43. O passo base está incorreto porque $n \neq 1$ para a soma mostrada. **45.** Seja $P(n)$ a afirmação "o plano está dividido em $n^2 - n + 2$ regiões por n círculos se cada dois destes círculos tiverem dois pontos em comum, mas quaisquer três deles não tiverem nenhum ponto em comum". *Passo base:* $P(1)$ é verdadeira porque um círculo divide o plano em $2 = 1^2 - 1 + 2$ regiões. *Passo de indução:* Suponha que $P(k)$ seja verdadeira, ou seja, k círculos com as propriedades especificadas dividem o plano em $k^2 - k + 2$ regiões. Suponha que seja adicionado um $(k + 1)$-ésimo círculo. Este círculo intercepta cada um dos outros k círculos em dois pontos, de modo que estes pontos de interseção formam $2k$ novos arcos, cada um dos quais divide uma antiga região. Logo, existem $2k$ divisões de regiões, o que mostra que existem $2k$ mais regiões do que existia anteriormente. Logo, $k + 1$ círculos satisfazendo as propriedades especificadas dividem o plano em $k^2 - k + 2 + 2k = (k^2 + 2k + 1) - (k + 1) + 2 = (k + 1)^2 - (k + 1) + 2$ regiões. **47.** Suponha que $\sqrt{2}$ seja racional. Então $\sqrt{2} = a/b$, em que a e b são inteiros positivos. Segue que o conjunto $S = \{n\ \sqrt{2}\ | n \in \mathbf{N}\} \cap \mathbf{N}$ é um conjunto não vazio de inteiros positivos, porque $b\ \sqrt{2} = a$ pertence a S. Seja t o menor elemento de S, que existe pela propriedade da boa ordenação. Então $t = s\sqrt{2}$ para algum inteiro s. Temos $t - s = s\ \sqrt{2} - s = s(\sqrt{2} - 1)$, de modo que $t - s$ é um inteiro positivo porque $\sqrt{2} > 1$. Logo, $t - s$ pertence a S. Isto é uma contradição, porque $t - s = s\ \sqrt{2} - s < s$. Logo, $\sqrt{2}$ é irracional. **49. a)** Seja $d = \text{mdc}(a_1, a_2, ..., a_n)$. Então d é um divisor de cada a_i e, deste modo, deve ser um divisor de $\text{mdc}(a_{n-1}, a_n)$. Logo, d é um divisor comum de $a_1, a_2, ..., a_{n-2}$, e $\text{mdc}(a_{n-1}, a_n)$. Para mostrar que ele é o maior divisor comum destes números, suponha que c seja um divisor comum deles. Então, c é um divisor de a_i para $i = 1, 2, ..., n - 2$ e um divisor de $\text{mdc}(a_{n-1}, a_n)$, de modo que ele é um divisor de a_{n-1} e a_n. Logo, c é um divisor comum de $a_1, a_2, ..., a_{n-1}$ e a_n. Logo, ele é um divisor de d, o maior divisor comum de $a_1, a_2, ..., a_n$. Segue que d é o maior divisor comum, como afirmado. **b)** Se $n = 2$, aplique o algoritmo de Euclides. Caso contrário, aplique o algoritmo de Euclides a a_{n-1} e a_n, obtendo $d = \text{mdc}(a_{n-1}, a_n)$, e então aplique o algoritmo recursivamente a $a_1, a_2, ..., a_{n-2}, d$. **51.** $f(n) = n^2$. Seja $P(n)$ a afirmação "$f(n) = n^2$". *Passo base:* $P(1)$ é verdadeira porque $f(1) = 1 = 1^2$, o que segue da definição de f. *Passo de indução:* Suponha que $f(n) = n^2$. Então, $f(n + 1) = f((n + 1) - 1) + 2(n + 1) - 1 = f(n) + 2n + 1 = n^2 + 2n + 1 = (n + 1)^2$. **53. a)** λ, 0, 1, 00, 01, 11, 000, 001, 011, 111, 0000, 0001, 0011, 0111, 1111, 00000, 00001, 00011, 00111, 01111, 11111. **b)** $S = \{\alpha\beta\ |\ \alpha$ é uma seqüência de m 0s e β é uma seqüência de n 1s, $m \geq 0, n \geq 0\}$ **55.** Aplique o primeiro passo recursivo a λ para obter $() \in B$. Aplique o segundo passo recursivo a esta seqüência para obter $()() \in B$. Aplique o primeiro passo recursivo a esta seqüência para obter $(()()) \in B$. Pelo Exercício 58, $(())$ não está em B porque o número de parênteses à esquerda não é igual ao número de parênteses à direita. **57.** λ, $()$, $(())$, $()()$ **59. a)** 0 **b)** -2 **c)** 2 **d)** 0

61. **procedure** *gerar*(n: inteiro não negativo)

if n é ímpar **then**

begin

$S := S(n - 1)$; $T := T(n - 1)$

end
else if $n = 0$ **then**
begin
$S := \emptyset$; $T := \{\lambda\}$
end
else
begin
$T_1 := T(n-2)$; $S_1 := S(n-2)$
$T := T_1 \cup \{(x) \mid x \in T_1 \cup S_1$ e $l(x) = n-2\}$
$S := S_1 \cup \{xy \mid x \in T_1$ e $y \in T_1 \cup S_1$
e $l(xy) = n\}$
end $\{T \cup S$ é o conjunto de seqüências balanceadas de comprimento no máximo $n\}$

63. Se $x \leq y$ inicialmente, então $x := y$ não é executado, de modo que $x \leq y$ é uma afirmação final verdadeira. Se $x > y$ inicialmente, então $x := y$ é executado, de modo que $x \leq y$ é novamente uma afirmação final verdadeira.

65. **procedure** *contazeros* ($a_1, a_2, ..., a_n$: lista de inteiros)
if $n = 1$ **then**
if $a_1 = 0$ **then** *contazeros* $(a_1, a_2, ..., a_n) := 1$
else *contazeros* $(a_1, a_2, ..., a_n) := 0$
else
if $a_n = 0$ **then** *contazeros* $(a_1, a_2, ..., a_n)$:= *contazeros* $(a_1, a_2, ..., a_{n-1}) + 1$
else *contazeros* $(a_1, a_2, ..., a_n)$:= *contazeros* $(a_1, a_2, ..., a_{n-1})$

67. Demonstraremos que $a(n)$ é um número natural e $a(n) \leq n$. Isto é verdadeiro para o caso base $n = 0$ porque $a(0) = 0$. Agora, suponha que $a(n-1)$ seja um número natural e $a(n-1) \leq n-1$. Então $a(a(n-1))$ é a aplicado a um número natural menor que ou igual a $n-1$. Logo, $a(a(n-1))$ também é um número natural menor que ou igual a $n-1$. Portanto, $n - a(a(n-1))$ é n menos algum número natural menor que ou igual a $n-1$, que é um número natural menor que ou igual a n. **69.** Do Exercício 68, $a(n) = \lfloor (n+1)\mu \rfloor$ e $a(n-1) = \lfloor n\mu \rfloor$. Como $\mu < 1$, estes dois valores são iguais ou diferem por 1. Primeiro, suponha que $\mu n - \lfloor \mu n \rfloor < 1 - \mu$. Isto é equivalente a $\mu(n+1) < 1 + \lfloor \mu n \rfloor$. Se isto for verdadeiro, então $\lfloor \mu(n+1) \rfloor = \lfloor \mu n \rfloor$. Por outro lado, se $\mu n - \lfloor \mu n \rfloor \geq 1 - \mu$, então $\mu(n+1) \geq 1 + \lfloor \mu n \rfloor$, de modo que $\lfloor \mu(n+1) \rfloor = \lfloor \mu n \rfloor + 1$, como desejado. **71.** $f(0) = 1$, $m(0) = 0$; $f(1) = 1$, $m(1) = 0$; $f(2) = 2$, $m(2) = 1$; $f(3) = 2$, $m(3) = 2$; $f(4) = 3$, $m(4) = 2$; $f(5) = 3$, $m(5) = 3$; $f(6) = 4$, $m(6) = 4$; $f(7) = 5$, $m(7) = 4$; $f(8) = 5$, $m(8) = 5$; $f(9) = 6$, $m(9) = 6$. **73.** A última ocorrência de n é na posição para a qual o número total de 1s, 2s, ..., ns todos juntos é aquele número de posição. Mas, como a_k é o número de ocorrências de k, isto é simplesmente $\sum_{k=1}^{n} a_k$, como desejado. Como $f(n)$ é a soma dos primeiros n termos da seqüência, $f(f(n))$ é a soma dos primeiros $f(n)$ termos da seqüência. Mas, como $f(n)$ é o último termo cujo valor é n, isto significa que a soma é a soma de todos os termos da seqüência cujos valores são no máximo n. Como existem a_k termos desta seqüência cujo valor é k, esta soma é $\sum_{k=1}^{n} k \cdot a_k$, como desejado.

CAPÍTULO 5

Seção 5.1

1. a) 5850 **b)** 343 **3. a)** 4^{10} **b)** 5^{10} **5.** 42 **7.** 26^3 **9.** 676 **11.** 2^8 **13.** $n+1$ (contando a seqüência vazia) **15.** 475255 (contando a seqüência vazia) **17.** 1321368961 **19. a)** Sete: 56, 63, 70, 77, 84, 91, 98 **b)** Cinco: 55, 66, 77, 88, 99 **c)** Um: 77 **21. a)** 12 8 **b)** 450 **c)** 9 **d)** 675 **e)** 450 **f)** 450 **g)** 225 **h)** 75 **23. a)** 990 **b)** 500 **c)** 27 **25.** 3^{50} **27.** 52457600 **29.** 20077200 **31. a)** 37822859361 **b)** 8204716800 **c)** 40159050880 **d)** 12113640000 **e)** 171004205215 **f)** 72043541640 **g)** 6230721635 **h)** 223149655 **33. a)** 0 **b)** 120 **c)** 720 **d)** 2520 **35. a)** 2 se $n = 1, 2$ se $n = 2$, 0 se $n \geq 3$ **b)** 2^{n-2} para $n > 1$; 1 se $n = 1$ **c)** $2(n-1)$ **37.** $(n+1)^m$ **39.** Se n for par, $2^{n/2}$; se n for ímpar, $2^{(n+1)/2}$ **41. a)** 240 **b)** 480 **c)** 360 **43.** 352 **45.** 147 **47.** 33 **49. a)** 9920671339261325541376 $\approx$ 9,9 $\times$ 10^{21} **b)** 6641514961387068437760 $\approx$ 6,6 $\times$ 10^{21} **c)** 9920671, 339.261325541376 segundos, o que é cerca de 314.000 anos. **51.** 7104000000000 **53.** 18 **55.** 17 **57.** 22 **59.** Seja $P(m)$ a regra da soma para m tarefas. Para o caso base, considere $m = 2$. Isto é simplesmente a regra da soma para duas tarefas. Suponha agora que $P(m)$ seja verdadeira. Considere $m+1$ tarefas, $T_1, T_2, ..., T_m, T_{m+1}$, que podem ser feitas de $n_1, n_2, ..., n_m, n_{m+1}$ maneiras, respectivamente, de modo que nenhum par destas tarefas possa ser feito ao mesmo tempo. Para realizar uma dessas tarefas, podemos ou fazer uma entre as primeiras m delas ou fazer a tarefa T_{m+1}. Pela regra da soma para duas tarefas, o número de maneiras de fazer isso é a soma do número de maneiras de fazer uma das primeiras m tarefas, mais n_{m+1}. Pela hipótese de indução, isto é $n_1 + n_2 + \ldots + n_m + n_{m+1}$, como desejado. **61.** $n(n-3)/2$

Seção 5.2

1. Como há seis aulas e apenas cinco dias úteis na semana, o princípio da casa dos pombos mostra que pelo menos duas aulas devem ocorrer no mesmo dia. **3. a)** 3 **b)** 14 **5.** Como existem quatro restos possíveis quando um inteiro é dividido por 4, o princípio da casa dos pombos implica que dados cinco inteiros, pelo menos dois têm o mesmo resto. **7.** Sejam $a, a+1, ..., a+n-1$ inteiros na seqüência. Os inteiros $(a+i)$ **mod** n, $i = 0, 1, 2, ..., n-1$, são distintos, pois $0 < (a+j) - (a+k) < n$ sempre que $0 \leq k < j \leq n-1$. Como existem n valores possíveis para $(a+i)$ **mod** n e existem n inteiros diferentes no conjunto, cada um destes valores é assumido exatamente uma vez. Segue que existe exatamente um inteiro na seqüência que é divisível por n. **9.** 4951 **11.** O ponto médio do segmento que liga (a, b, c) e (d, e, f) é $((a+d)/2, (b+e)/2, (c+f)/2)$. Ele tem coeficientes inteiros se e somente se a e d tiverem a mesma paridade, b e e tiverem

a mesma paridade e c e f tiverem a mesma paridade. Como existem oito triplas possíveis de paridade [tal como (*par*, *ímpar*, *par*)], pelo princípio da casa dos pombos pelo menos dois dos nove pontos têm a mesma tripla de paridade. O ponto médio do segmento que liga dois desses pontos tem coeficientes inteiros. **13. a)** Agrupe os primeiros oito inteiros positivos em quatro subconjuntos de dois inteiros cada, de modo que os inteiros de cada subconjunto somem 9: {1, 8}, {2, 7}, {3, 6} e {4, 5}. Se cinco inteiros forem escolhidos dos primeiros oito inteiros positivos, pelo princípio da casa dos pombos, pelo menos dois deles vêm do mesmo subconjunto. Dois desses inteiros têm a soma 9, como desejado. **b)** Não. Considere {1, 2, 3, 4}, por exemplo. **15.** 4 **17.** 21251 **19. a)** Se existissem menos do que 9 calouros, menos do que 9 veteranos e menos do que 9 alunos juniores na classe, não haveria mais de 8 de cada um destes anos, para um total de no máximo 24 estudantes, contradizendo o fato de haver 25 estudantes no curso. **b)** Se houver menos de 3 calouros, menos de 19 veteranos e menos do que 5 alunos juniores, então haveria no máximo 2 calouros, no máximo 18 veteranos e no máximo 4 alunos juniores, para um total de no máximo 24 estudantes. Isto contradiz o fato de haver 25 estudantes no curso. **21.** 4, 3, 2, 1, 8, 7, 6, 5, 12, 11, 10, 9, 16, 15, 14, 13.

23. **procedure** *long*(a_1, ..., a_n: inteiros positivos)
{primeiro encontra a subseqüência crescente mais comprida}
max := 0; *set* := 00 ...00 {n bits}
for $i := 1$ **to** 2^n
begin
last := 0; *count* := 0, *OK* := **true**
for $j := 1$ **to** n
begin
if *set*(j) = 1 **then**
begin
if $a_j >$ *last* **then** *last* := a_j
count := *count* + 1
end
else *OK* := **false**
end
if *count* > *max* **then**
begin
max := *count*
best := *set*
end
set := *set* + 1 (adição binária)
end {*max* é o comprimento e *best* indica a seqüência}
{repita para subseqüências decrescentes com as únicas mudanças sendo $a_j <$ *last* em vez de $a_j >$ *last* e *last* := ∞ em vez de *last* := 0}

25. Por simetria, precisamos demonstrar apenas a primeira afirmação. Seja A uma das pessoas. Ou A tem pelo menos quatro amigos ou A tem pelo menos seis inimigos entre as outras nove pessoas (pois $3 + 5 < 9$). Suponha, no primeiro caso, que B, C, D e E são todos amigos de A. Se quaisquer dois deles forem amigos entre si, então encontramos três amigos mútuos. Caso contrário, {B, C, D, E} é um conjunto de quatro inimigos mútuos. No segundo caso, seja {B, C, D, E, F, G} um conjunto de inimigos de A. Pelo Exemplo 11, entre B, C, D, E, F e G existem ou três amigos mútuos ou três inimigos mútuos, que formam, com A, um conjunto de quatro inimigos mútuos. **27.** Precisamos mostrar duas coisas: que, se tivermos um grupo de n pessoas, então entre elas devemos encontrar ou um par de amigos ou um subconjunto de n delas que são inimigas mútuas; e que existe um grupo de $n - 1$ pessoas para as quais isto não é possível. Para a primeira afirmação, se existir qualquer par de amigos, então a condição é satisfeita, e, se não, então todo par de pessoas são inimigas, de modo que a segunda condição é satisfeita. Para a segunda afirmação, se tivermos um grupo de $n - 1$ pessoas que são todas inimigas umas das outras, então não existe nem um par de amigos nem um subconjunto n delas que sejam todas inimigas entre si. **29.** Existem 6432816 possibilidades para as três iniciais e um aniversário. Assim, pelo princípio da casa dos pombos generalizado, existem pelo menos $\lceil 36000000/6432816 \rceil = 6$ pessoas que compartilham as mesmas iniciais e aniversários. **31.** 18 **33.** Como existem seis computadores, o número de outros computadores aos quais um computador está conectado é um inteiro entre 0 e 5, inclusive. Entretanto, 0 e 5 não podem ambos ocorrer. Para ver isso, observe que, se algum computador não estiver conectado a nenhum outro, então nenhum computador está conectado aos outros cinco, e, se algum computador estiver conectado a todos os outros cinco, então nenhum computador deixará de estar conectado a algum outro. Assim, pelo princípio da casa dos pombos, como existem no máximo cinco possibilidades para o número de computadores aos quais um computador está conectado, existe pelo menos dois computadores no conjunto de seis computadores conectados ao mesmo número de outros computadores. **35.** Rotule os computadores de C_1 até C_{100} e rotule as impressoras de P_1 até P_{20}. Se conectarmos C_k a P_k para $k = 1$, 2, ... , 20 e conectarmos cada um dos computadores C_{21} até C_{100} a *todas* as impressoras, então teremos usado um total de $20 + 80 \cdot 20 = 1620$ cabos. Claramente, isto é suficiente, pois, se os computadores C_1 até C_{20} precisarem de impressoras, então eles podem usar aquelas com os mesmos subscritos, e, se quaisquer computadores com subscritos mais altos precisarem de uma impressora em vez de um ou mais deles, então eles podem usar as impressoras que não estiverem sendo usadas, pois eles estão conectados a todas as impressoras. Agora, precisamos mostrar que 1619 cabos não são suficientes. Como existem 1619 cabos e 20 impressoras, o número médio de computadores por impressora é 1619/20, que é menos de 81. Portanto, alguma impressora deve estar conectada a menos de 81 computadores. Isto significa que ela está conectada a 80 ou menos computadores, de modo que existem 20 computadores que não estão conectados a ela. Se todos estes 20 computadores precisarem de uma impressora simultaneamente, então eles estariam sem sorte, pois estariam conectados a, no máximo, 19 outras. **37.** Seja a_i o número de competições completadas até a hora i. Então, $1 \leq a_1 < a_2 < \cdots < a_{75} \leq 125$. Além disso, $25 \leq a_1 + 24 < a_2 + 24 < \cdots < a_{75} + 24 \leq 149$. Existem 150 números a_1, ..., a_{75}, $a_1 + 24$, ... , $a_{75} + 24$. Pelo princípio da casa dos pombos, pelo menos dois são iguais.

Como todos os a_is são distintos e todos os $(a_i + 24)$s são distintos, segue que $a_i = a_j + 24$ para algum $i > j$. Logo, no período da $(j + 1)$-ésima até a i-ésima hora, ocorreram exatamente 24 competições. **39.** Use o princípio da casa dos pombos generalizado, colocando os $|S|$ objetos $f(s)$ para $s \in S$ em $|T|$ caixas, uma para cada elemento de T. **41.** Faça d_j igual a $jx - N(jx)$, em que $N(jx)$ é o inteiro mais próximo de jx para $1 \leq j \leq n$. Cada d_j é um número irracional entre $-1/2$ e $1/2$. Vamos supor que n seja par; o caso em que n é ímpar é mais complicado. Considere os n intervalos $\{x \mid j/n < x < (j+1)/n\}$, $\{x \mid -(j+1)/n < x < -j/n\}$ para $j = 0, 1, \ldots, (n/2) - 1$. Se d_j pertencer ao intervalo $\{x \mid 0 < x < 1/n\}$ ou ao intervalo $\{x \mid -1/n < x < 0\}$ para algum j, teremos acabado. Se não, como existem $n - 2$ intervalos e n números d_j, o princípio da casa dos pombos nos diz que existe um intervalo $\{x \mid (k-1)/n < x < k/n\}$ que contém d_r e d_s com $r < s$. A demonstração pode ser concluída mostrando que $(s - r)x$ está a menos de $1/n$ de seu inteiro mais próximo. **43. a)** Suponha que $i_k \leq n$ para todo k. Então, pelo princípio da casa dos pombos generalizado, pelo menos $\lceil (n^2+1)/n \rceil = n + 1$ dos números $i_1, i_2, \ldots, i_{n^2+1}$ são iguais. **b)** Se $a_{k_j} < a_{k_{j+1}}$, então a subseqüência que consiste dos a_{k_j} seguidos por uma subseqüência crescente de comprimento $i_{k_{j+1}}$ começando em $a_{k_{j+1}}$ contradiz o fato de que $i_{k_j} = i_{k_{j+1}}$. Logo, $a_{k_j} > a_{k_{j+1}}$. **c)** Se não existir nenhuma subseqüência crescente de comprimento maior do que n, então as partes (a) e (b) se aplicam. Portanto, temos $a_{k_{n+1}} > a_{k_n} > \cdots > a_{k_2} > a_{k_1}$, uma seqüência decrescente de comprimento $n + 1$.

Seção 5.3

1. *abc*, *acb*, *bac*, *bca*, *cab*, *cba* **3.** 720 **5. a)** 120 **b)** 72 0 **c)** 8 **d)** 6720 **e)** 40 320 **f)** 3 628 800 **7.** 15 120 **9.** 1320 **11. a)** 210 **b)** 386 **c)** 848 **d)** 252 **13.** $2(n!)^2$ **15.** 65 780 **17.** $2^{100} - 5051$ **19. a)** 1024 **b)** 45 **c)** 176 **d)** 252 **21. a)** 120 **b)** 24 **c)** 120 **d)** 24 **e)** 6 **f)** 0 **23.** 609 638 400 **25. a)** 94 109 400 **b)** 941 094 **c)** 3 764 376 **d)** 90 345 024 **e)** 114 072 **f)** 2328 **g)** 24 **h)** 79 727 040 **i)** 3 764 376 **j)** 109 440 **27. a)** 12 650 **b)** 303 600 **29. a)** 37 927 **b)** 18 915 **31. a)** 122 523 030 **b)** 72 930 375 **c)** 223 149 655 **d)** 100 626 625 **33.** 54 600 **35.** 45 **37.** 912 **39.** 11 232 000 **41.** 13 **43.** 873

Seção 5.4

1. $x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4$ **3.** $x^6 + 6x^5y + 15x^4y^2 + 20x^3y^3 + 15x^2y^4 + 6xy^5 + y^6$ **5.** 101 **7.** $-2^{10}\binom{19}{9} = -94\,595\,072$ **9.** $-2^{101}3^{99}\binom{200}{99}$ **11.** $(-1)^{(200-k)/3}\binom{100}{(200-k)/3}$ se $k \equiv 2 \pmod 3$ e $-100 \leq k \leq 200$; 0 caso contrário **13.** 1 9 36 84 126 126 84 36 9 1 **15.** A soma de *todos* os números positivos $\binom{n}{k}$, com k variando de 0 a n, é 2^n, de modo que cada um deles não é maior que sua soma. **17.** $\binom{n}{k} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k(k-1)(k-2)\cdots 2} \leq \frac{n \cdot n \cdot \cdots \cdot n}{2 \cdot 2 \cdot \cdots \cdot 2} = n^k/2^{k-1}$ **19.** $\binom{n}{k-1} + \binom{n}{k} = \frac{n!}{(k-1)!(n-k+1)!} + \frac{n!}{k!(n-k)!} = \frac{n!}{k!(n-k+1)!} \cdot [k + (n-k+1)] = \frac{(n+1)!}{k!(n+1-k)!} = \binom{n+1}{k}$ **21. a)** Mostramos que cada lado conta o número de maneiras de selecionar um subconjunto com k elementos a partir de um conjunto com n elementos e um elemento distinto naquele conjunto. Para o lado esquerdo, primeiro escolha o conjunto de k elementos (isto pode ser feito de $\binom{n}{k}$ maneiras) e a seguir escolha um dos k elementos neste subconjunto como o elemento distinto (isto pode ser feito de k maneiras). Para o lado direito, primeiro escolha o elemento distinto do conjunto todo de n elementos (isto pode ser feito de n maneiras) e a seguir escolha os $k - 1$ elementos restantes do subconjunto dos $n - 1$ elementos restantes do conjunto (isto pode ser feito de $\binom{n-1}{k-1}$ maneiras). **b)** $k\binom{n}{k} = k \cdot \frac{n!}{k!(n-k)!} = \frac{n \cdot (n-1)!}{(k-1)!(n-k)!} = n\binom{n-1}{k-1}$ **23.** $\binom{n+1}{k} = \frac{(n+1)!}{k!(n+1-k)!} = \frac{(k+1)}{k}\frac{n!}{(k-1)!(n-(k-1))!} = (n+1)\binom{n}{k-1}/k$. Esta identidade junto com $\binom{n}{0} = 1$ dá uma definição recursiva. **25.** $\binom{2n}{n+1} + \binom{2n}{n} = \binom{2n+1}{n+1} = \frac{1}{2}\left[\binom{2n+1}{n+1} + \binom{2n+1}{n+1}\right] = \frac{1}{2}\left[\binom{2n+1}{n+1} + \binom{2n+1}{n}\right] = \frac{1}{2}\binom{2n+2}{n+1}$ **27.** a) $\binom{n+r+1}{r}$ conta o número de maneiras de escolher uma seqüência de r 0s e $n + 1$ 1s escolhendo as posições dos 0s. Alternativamente, suponha que o $(j + 1)$-ésimo termo seja o último termo igual a 1, de modo que $n \leq j \leq n + r$. Uma vez que tenhamos determinado onde o último 1 está, decidimos onde os 0s devem ser colocados nos j espaços antes do último 1. Existem n 1s e $j - n$ 0s neste intervalo. Pela regra da soma, segue que existem $\sum_{j=n}^{n+r}\binom{j}{j-n} = \sum_{k=0}^{r}\binom{n+k}{k}$ maneiras de fazer isto. **b)** Seja $P(r)$ a afirmação a ser demonstrada. O passo base é a equação $\binom{n}{0} = \binom{n+1}{0}$, que é simplesmente $1 = 1$. Suponha que $P(r)$ seja verdadeira. Então $\sum_{k=0}^{r+1}\binom{n+k}{k} = \sum_{k=0}^{r}\binom{n+k}{k} + \binom{n+r+1}{r+1} = \binom{n+r+1}{r} + \binom{n+r+1}{r+1} = \binom{n+r+2}{r+1}$, usando a hipótese de indução e a identidade de Pascal. **29.** Podemos escolher o líder primeiro de n maneiras diferentes. Podemos então escolher o resto do comitê de 2^{n-1} maneiras. Assim, existem $n2^{n-1}$ maneiras de escolher um comitê e seu líder. Entretanto, o número de maneiras de escolher um comitê com k pessoas é $\binom{n}{k}$. Uma vez que tenhamos escolhido um comitê com k pessoas, existem k maneiras de escolher seu líder. Assim, existem $\sum_{k=1}^{n} k\binom{n}{k}$ maneiras de escolher o comitê e seu líder. Assim, $\sum_{k=1}^{n} k\binom{n}{k} = n2^{n-1}$. **31.** Suponha que o conjunto tenha n elementos. Do Corolário 2 temos $\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \cdots + (-1)^n\binom{n}{n} = 0$. Segue que $\binom{n}{0} + \binom{n}{2} + \binom{n}{4} + \cdots = \binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \cdots$. O lado esquerdo dá o número de subconjuntos com um número par de elementos, e o lado direito dá o número de subconjuntos com um número ímpar de elementos. **33. a)** Um caminho do tipo desejado consiste em m movimentos para a direita e n movimentos para cima. Cada caminho destes pode ser representado por uma cadeia de bits de comprimento $m + n$, com m 0s e n 1s, em que um 0 representa um movimento para a direita e um 1, um movimento para cima. **b)** O número de cadeias de bits de comprimento $m + n$ com exatamente n 1s é igual a $\binom{m+n}{n} = \binom{m+n}{m}$, pois tal cadeia é determinada pela especificação das posições dos n 1s ou pela especificação das posições dos m 0s. **35.** Pelo Exercício 33, o número de caminhos de

comprimento n do tipo descrito naquele exercício é igual a 2^n, o número de cadeias de bits de comprimento n. Por outro lado, um caminho de comprimento n do tipo descrito no Exercício 33 deve terminar em um ponto que tem n como a soma de suas coordenadas, digamos $(n-k, k)$ para algum k entre 0 e n, inclusive. Pelo Exercício 33, o número de tais caminhos que terminam em $(n-k, k)$ é igual a $\binom{n-k+k}{k} = \binom{n}{k}$. Assim, $\sum_{k=0}^{n}\binom{n}{k} = 2^n$. **37.** Pelo Exercício 33, o número de caminhos de (0, 0) a $(n+1, r)$ do tipo descrito naquele exercício é igual a $\binom{n+r+1}{r}$. Mas tal caminho começa indo j passos verticalmente para algum j com $0 \le j \le r$. O número de tais caminhos que começam com j passos verticais é igual ao número de caminhos do tipo descrito no Exercício 33, que vão de $(1, j)$ a $(n+1, r)$. Isto é o mesmo que o número de tais caminhos que vão de (0, 0) a $(n, r-j)$, o que, pelo Exercício 33, é igual a $\binom{n+r-j}{r-j}$. Como $\sum_{j=0}^{r}\binom{n+r-j}{r-j} = \sum_{k=0}^{r}\binom{n+k}{k}$, segue que $\sum_{k=1}^{r}\binom{n+k}{k} = \binom{n+r+1}{r}$. **39. a)** $\binom{n+1}{2}$ **b)** $\binom{n+2}{3}$ **c)** $\binom{2n-2}{n-1}$ **d)** $\binom{\lfloor (n-1)/2 \rfloor}{n-1}$ **e)** Maior elemento ímpar na n-ésima linha do triângulo de Pascal. **f)** $\binom{3n-3}{n-1}$

Seção 5.5

1. 243 **3.** 26^6 **5.** 125 **7.** 35 **9. a)** 1716 **b)** 50388 **c)** 2629575 **d)** 330 **e)** 9724 **11.** 9 **13.** 4504501 **15. a)** 10626 **b)** 1365 **c)** 11649 **d)** 106 **17.** 2520 **19.** 302702400 **21.** 3003 **23.** 7484400 **25.** 30492 **27.** $C(59\,50)$ **29.** 35 **31.** 83160 **33.** 63 **35.** 19635 **37.** 210 **39.** 27720 **41.** $52!/(7!^5 17!)$ **43.** Aproximadamente $6{,}5 \times 10^{32}$ **45. a)** $C(k+n-1, n)$ **b)** $(k+n-1)!/(k-1)!$ **47.** Existem $C(n, n_1)$ maneiras de escolher n_1 objetos para a primeira caixa. Uma vez que estes objetos tenham sido escolhidos, existem $C(n-n_1, n_2)$ maneiras de escolher objetos para a segunda caixa. Analogamente, existem $C(n-n_1-n_2, n_3)$ maneiras de escolher objetos para a terceira caixa. Continue desta maneira até que existam $C(n-n_1-n_2-\cdots-n_{k-1}, n_k) = C(n_k, n_k) = 1$ maneiras de escolher objetos para a última caixa (já que $n_1+n_2+\cdots+n_k = n$). Pela regra do produto, o número de maneiras de fazer a associação toda é $C(n, n_1)C(n-n_1, n_2)C(n-n_1-n_2, n_3)\cdots C(n-n_1-n_2-\cdots-n_{k-1}, n_k)$, que é igual a $n!/(n_1!n_2!\cdots n_k!)$, como mostra uma fácil simplificação. **49. a)** Como $x_1 \le x_2 \le \cdots \le x_r$, segue que $x_1 + 0 < x_2 + 1 < \cdots < x_r + r - 1$. As desigualdades são estritas, pois $x_j + j - 1 < x_{j+1} + j$ contanto que $x_j \le x_{j+1}$. Como $1 \le x_j \le n+r-1$, esta seqüência é formada por r elementos distintos de T. **b)** Suponha que $1 \le x_1 < x_2 < \cdots < x_r \le n+r-1$. Seja $y_k = x_k - (k-1)$. Então, não é difícil ver que $y_k \le y_{k+1}$ para $k = 1, 2, \ldots, r-1$ e que $1 \le y_k \le n$ para $k = 1, 2, \ldots r$. Segue que $\{y_1, y_2, \ldots, y_r\}$ é uma r-combinação com repetições permitidas de S. **c)** Das partes (a) e (b) segue que existe uma correspondência bijetora das r-combinações com repetições permitidas de S e r-combinações de T, um conjunto com $n+r-1$ elementos. Concluímos que existem $C(n+r-1, r)$ r-combinações com repetições permitidas de S. **51.** 65 **53.** 65 **55.** 2 **57.** 3 **59. a)** 150 **b)** 25 **c)** 6 **d)** 2 **61.** 90720 **63.** Os termos na expansão são da forma $x_1^{n_1}x_2^{n_2}\cdots x_m^{n_m}$, em que $n_1+n_2+\cdots+n_m = n$. Tal termo aparece da escolha de x_1 em n_1 fatores, de x_2 em n_2 fatores, ..., e de x_m em n_m fatores. Isto pode ser feito em $C(n; n_1, n_2, \ldots, n_m)$ maneiras, pois uma escolha é uma permutação de n_1 rótulos "1", n_2 rótulos "2", ..., e n_m rótulos "m". **65.** 2520

Seção 5.6

1. 14532, 15432, 21345, 23451, 23514, 31452, 31542, 43521, 45213, 45321 **3. a)** 2134 **b)** 54132 **c)** 12534 **d)** 45312 **e)** 6714253 **f)** 31542678 **5.** 1234, 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312, 4321 **7.** {1, 2, 3}, {1, 2, 4}, {1, 2, 5}, {1, 3, 4}, {1, 3, 5}, {1, 4, 5}, {2, 3, 4}, {2, 3, 5}, {2, 4, 5}, {3, 4, 5} **9.** A cadeia de bits que representa a próxima r-combinação maior deve diferir da cadeia de bits que representa a original na posição i, pois as posições $i+1, \ldots, r$ estão ocupadas pelos maiores números possíveis. Além disso, $a_i + 1$ é o menor número possível que podemos colocar na posição i se queremos uma combinação maior do que a original. Então, $a_i+2, \ldots, a_i+r-i+1$ são os menores números permitidos para as posições $i+1$ a r. Logo, produzimos a próxima r-combinação. **11.** 123, 132, 213, 231, 312, 321, 124, 142, 214, 241, 412, 421, 125, 152, 215, 251, 512, 521, 134, 143, 314, 341, 413, 431, 135, 153, 315, 351, 513, 531, 145, 154, 415, 451, 514, 541, 234, 243, 324, 342, 423, 432, 235, 253, 325, 352, 523, 532, 245, 254, 425, 452, 524, 542, 345, 354, 435, 453, 534, 543 **13.** Mostraremos que é uma bijeção mostrando que tem uma inversa. Dado um inteiro positivo menor do que $n!$, sejam $a_1, a_2, \ldots, a_{n-1}$ seus algarismos de Cantor. Ponha n na posição $n - a_{n-1}$; então, claramente, a_{n-1} é o número de inteiros menor que n que segue n na permutação. Então, coloque $n-1$ na posição livre $(n-1) - a_{n-2}$, em que enumeramos as posições livres por 1, 2, ..., $n-1$ (excluindo a posição que n já está ocupando). Continue até que 1 seja colocado na única posição livre restante. Como construímos uma inversa, a correspondência é uma bijeção.

15. **procedure** *permutacao de Cantor* (n, i: inteiros com $n \ge 1$ e $0 \le i < n!$)
```
x := n
for j := 1 to n
  p_j := 0
for k := 1 to n − 1
begin
  c := ⌊x/(n − k)!⌋; x := x − c(n − k)!; h := n
  while p_h ≠ 0
    h := h − 1
  for j := 1 to c
  begin
    h := h − 1
    while p_h ≠ 0
      h := h − 1
  end
  p_h := n − k + 1
end
h := 1
while p_h ≠ 0
```

$h := h + 1$
$p_h := 1$
{$p_1 p_2 \cdots p_n$ é a permutação correspondente a i}

Exercícios Complementares

1. **a)** 151 200 **b)** 1 000 000 **c)** 210 **d)** 5005 **3.** 3^{100} **5.** 24 600 **7.** **a)** 4060 **b)** 2688 **c)** 25 009 600 **9.** **a)** 192 **b)** 301 **c)** 300 **d)** 300 **11.** 639 **13.** A soma máxima possível é 240, e a soma mínima possível é 15. Assim, o número possível de somas é 226. Como existem 252 subconjuntos com cinco elementos de um conjunto com 10 elementos, pelo princípio da casa dos pombos, segue que pelo menos dois têm a mesma soma. **15.** **a)** 50 **b)** 50 **c)** 14 **d)** 5 **17.** Sejam $a_1, a_2, \ldots, a_m$ os inteiros e sejam $d_i = \sum_{j=1}^{i} a_j$. Se $d_i \equiv 0 (\text{mod } m)$ para algum i, teremos acabado. Caso contrário d_1 **mod** m, d_2 **mod** m, ..., d_m **mod** m são m inteiros com valores em $\{1, 2, \ldots, m-1\}$. Pelo princípio da casa dos pombos, $d_k = d_l$ para algum $1 \leq k < l \leq m$. Então $\sum_{j=k+1}^{l} a_j = d_l - d_k \equiv 0 (\text{mod } m)$. **19.** A expansão decimal do número racional a/b pode ser obtida pela divisão de a por b, em que a é escrito com uma vírgula decimal seguida por uma seqüência arbitrariamente longa de 0s. O passo básico é encontrar o próximo algarismo do quociente, a saber, $\lfloor r/b \rfloor$, em que r é o resto com o próximo algarismo do dividendo abaixado. O resto corrente é obtido a partir do resto anterior subtraindo b vezes o algarismo anterior do quociente. Eventualmente, o dividendo não tem nada, exceto 0s para abaixar. Além disso, existem apenas b restos possíveis. Assim, em algum ponto, pelo princípio da casa dos pombos, teremos a mesma situação que apareceu anteriormente. Desse ponto em diante, o cálculo deve seguir o mesmo padrão. Em particular, o quociente se repetirá. **21.** **a)** 125 970 **b)** 20 **c)** 141 120 525 **d)** 141 120 505 **e)** 177 100 **f)** 141 078 021 **23.** **a)** 10 **b)** 8 **c)** 7 **25.** 3^n **27.** $C(n+2, r+1) = C(n+1, r+1) + C(n+1, r) = 2C(n+1, r+1) - C(n+1, r+1) + C(n+1, r) = 2C(n+1, r+1) - (C(n, r+1) + C(n, r)) + (C(n, r) + C(n, r-1)) = 2C(n+1, r+1) - C(n, r+1) + C(n, r-1)$ **29.** Substitua $x = 1$ e $y = 3$ no Teorema Binomial. **31.** $C(n+1, 5)$ **33.** 3 491 888 400 **35.** 5^{24} **37.** **a)** 45 **b)** 57 **c)** 12 **39.** **a)** 386 **b)** 56 **c)** 512 **41.** 0 se $n < m$; $C(n-1, n-m)$ se $n \geq m$ **43.** **a)** 15 625 **b)** 202 **c)** 210 **d)** 10 **45.** **procedure** *proxima permutacao* (n: inteiro positivo, a_1, a_2, ..., a_r: inteiros positivos que não excedam n com $a_1 a_2 \ldots a_r \neq nn \ldots n$)
$i := r$
while $a_i = n$
begin
$a_i := 1$
$i := i - 1$
end
$a_i := a_i + 1$
{$a_1 a_2 \ldots a_r$ é a próxima permutação na ordem lexicográfica}
47. Devemos mostrar que, se existirem $R(m, n-1) + R(m-1, n)$ pessoas em uma festa, então devem existir pelo menos m amigos mútuos ou n inimigos mútuos. Considere uma pessoa; vamos chamá-la de Jerry. Então, existem $R(m-1, n) + R(m, n-1) - 1$ outras pessoas na festa, e, pelo princípio da casa dos pombos, deve haver pelo menos $R(m-1, n)$ amigos de Jerry ou $R(m, n-1)$ inimigos de Jerry entre essas pessoas. Primeiro, vamos supor que existam $R(m-1, n)$ amigos de Jerry. Pela definição de R, temos a certeza de que entre essas pessoas encontraremos ou $m-1$ amigos mútuos ou n inimigos mútuos. No primeiro caso, estes $m-1$ amigos mútuos junto com Jerry são um conjunto de m amigos mútuos, e, no último caso, temos o conjunto pedido de n inimigos mútuos. A outra situação é parecida: suponha que existam $R(m, n-1)$ inimigos de Jerry; temos a certeza de encontrar entre eles ou m amigos mútuos ou $n-1$ inimigos mútuos. No primeiro caso, temos o conjunto pedido de m amigos mútuos, e, no último caso, estes $n-1$ inimigos mútuos junto com Jerry são um conjunto de n inimigos mútuos.

CAPÍTULO 6

Seção 6.1

1. 1/13 **3.** 1/2 **5.** 1/2 **7.** 1/64 **9.** 47/52 **11.** $1/C(52, 5)$ **13.** $1 - [C(48, 5)/C(52, 5)]$ **15.** $C(13, 2)C(4, 2)C(4, 2)C(44, 1)/C(52, 5)$ **17.** $10.240/C(52, 5)$ **19.** $1.302.540/C(52, 5)$ **21.** 1/64 **23.** 8/25 **25.** **a)** $1/C(50, 6) = 1/15.890.700$ **b)** $1/C(52, 6) = 1/20.358.520$ **c)** $1/C(56, 6) = 1/32.468.436$ **d)** $1/C(60, 6) = 1/50.063.860$ **27.** **a)** 139.128/319.865 **b)** 212.667/511.313 **c)** 151.340/386.529 **d)** 163.647/446.276 **29.** $1/C(100, 8)$ **31.** 3/100 **33.** **a)** 1/7.880.400 **b)** 1/8.000.000 **35.** **a)** 9/19 **b)** 81/361 **c)** 1/19 **d)** 1.889.568/2.476.099 **e)** 48/361 **37.** Três dados. **39.** A porta que o concorrente escolhe é escolhida aleatoriamente sem o conhecimento de onde o prêmio está, mas a porta escolhida pelo apresentador não é escolhida aleatoriamente, pois ele sempre evita abrir a porta com o prêmio. Isto torna qualquer argumento baseado em simetria inválido. **41.** **a)** 671/1296 **b)** $1 - 35^{24}/36^{24}$; não. **c)** Sim.

Seção 6.2

1. $p(T) = 1/4$, $p(H) = 3/4$ **3.** $p(1) = p(3) = p(5) = p(6) = 1/16$; $p(2) = p(4) = 3/8$ **5.** 9/49 **7.** **a)** 1/2 **b)** 1/2 **c)** 1/3 **d)** 1/4 **e)** 1/4 **9.** **a)** 1/26! **b)** 1/26 **c)** 1/2 **d)** 1/26 **e)** 1/650 **f)** 1/15.600 **11.** Claramente, $p(E \cup F) \geq p(E) = 0{,}7$. Além disso, $p(E \cup F) \leq 1$. Se aplicarmos o Teorema 2 da Seção 6.1, podemos reescrever isso como $p(E) + p(F) - p(E \cap F) \leq 1$, ou $0{,}7 + 0{,}5 - p(E \cap F) \leq 1$. Isolando $p(E \cap F)$ obtemos $p(E \cap F) \geq 0{,}2$. **13.** Como $p(E \cup F) = p(E) + p(F) - p(E \cap F)$ e $p(E \cup F) \leq 1$, segue que $1 \geq p(E) + p(F) - p(E \cap F)$. Desta desigualdade concluímos que $p(E) + p(F) \leq 1 + p(E \cap F)$. **15.** Usaremos indução matemática para demonstrar que esta desigualdade é válida para $n \geq 2$. Seja $P(n)$ a afirmação de que $p(\bigcup_{j=1}^{n} E_j) \leq \sum_{j=1}^{n} p(E_j)$. *Passo base:* $P(2)$ é verdadeira, pois $p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2) \leq p(E_1) + p(E_2)$. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Usando o caso base e a hipótese de indução, segue que $p(\bigcup_{j=1}^{k+1} E_j) \leq p(\bigcup_{j=1}^{k} E_j) + p(E_{k+1}) \leq \sum_{j=1}^{k+1} p(E_j)$. Isto

mostra que $P(k+1)$ é verdadeira, o que completa a demonstração por indução matemática. **17.** Como $E \cup \overline{E}$ é o espaço amostral inteiro S, o evento F pode ser dividido em dois eventos disjuntos: $F = S \cap F = (E \cup \overline{E}) \cap F = (E \cap F) \cup (\overline{E} \cap F)$, usando a propriedade distributiva. Portanto, $p(F) = p((E \cap F) \cup (\overline{E} \cap F)) = p(E \cap F) + p(\overline{E} \cap F)$, pois estes dois eventos são disjuntos. Subtraindo $p(E \cap F)$ de ambos os lados, usando o fato de que $p(E \cap F) = p(E) \cdot p(F)$ (a hipótese de que E e F são independentes), e fatorando, temos $p(F)[1 - p(E)] = p(\overline{E} \cap F)$. Como $1 - p(E) = p(\overline{E})$, isso diz que $p(\overline{E} \cap F) = p(\overline{E}) \cdot p(F)$, como desejado. **19. a)** $1/12$ **b)** $1 - \frac{11}{12} \cdot \frac{10}{12} \cdot \cdots \cdot \frac{13-n}{12}$ **c)** 5. **21.** 614 **23.** $1/4$ **25.** $3/8$ **27. a)** Não independentes. **b)** Não independentes. **c)** Não independentes. **29.** $3/16$. **31. a)** $1/32 = 0{,}03125$ **b)** $0{,}49^5 \approx 0{,}02825$ **c)** 0,03795012. **33. a)** $5/8$ **b)** 0,627649 **c)** 0,6431 **35. a)** p^n **b)** $1 - p^n$ **c)** $p^n + n \cdot p^{n-1} \cdot (1-p)$ **d)** $1 - [p^n + n \cdot p^{n-1} \cdot (1-p)]$. **37.** $p(\bigcup_{i=1}^{\infty} E_i)$ é a soma dos $p(s)$ para cada saída s em $\bigcup_{i=1}^{\infty} E_i$. Como os E_i s são dois a dois disjuntos, isto é a soma das probabilidades de todas as saídas em qualquer dos E_i s, que é o mesmo que $\sum_{i=1}^{\infty} p(E_i)$ é. (Podemos reordenar os somandos e ainda obter a mesma resposta, pois a série converge absolutamente.) **39. a)** $\overline{E} = \bigcup_{i=1}^{\binom{n}{k}} F_i$, de modo que a inequação dada agora segue da inequação de Boole (Exercício 15). **b)** A probabilidade de que um jogador qualquer que não esteja no j-ésimo conjunto ganhe de todos os k jogadores no j-ésimo conjunto é $(1/2)^k = 2^{-k}$. Portanto, a probabilidade de que este jogador não faça isto é $1 - 2^{-k}$, de modo que a probabilidade de que todos os $m - k$ jogadores que não estão no j-ésimo conjunto sejam incapazes de conseguir um placar perfeito contra todos no j-ésimo conjunto é $(1 - 2^{-k})^{m-k}$. Isto é precisamente $p(F_j)$. **c)** A primeira desigualdade segue imediatamente, pois todos os somandos são iguais e existem $\binom{m}{k}$ deles. Se esta probabilidade for menor que 1, então deve ser possível que $\overline{E}$ falhe, isto é, que E ocorra. Assim, existe um torneio que satisfaz as condições do problema contanto que a segunda desigualdade seja válida. **d)** $m \geq 21$ para $k = 2$ e $m \geq 91$ para $k = 3$

41. **procedure** *primo probabilistico* (n, k)
composto := **false**
$i := 0$
while *composto* = **false** e $i < k$
begin
$i := i + 1$
escolha b de modo uniformemente aleatório com $1 < b < n$
aplique o teste de Miller à base b
if n não satisfizer o teste **then** *composto* := **true**
end
if *composto* = **true then** imprima ("composto")
else imprima ("provavelmente primo")

Seção 6.3

OBSERVAÇÃO: Nas respostas da Seção 6.3, todas as probabilidades dadas na forma decimal são arredondadas para três casas decimais. **1.** $3/5$ **3.** $3/4$ **5.** 0,481 **7. a)** 0,999 **b)** 0,324 **9. a)** 0,740 **b)** 0,260 **c)** 0,002 **d)** 0,998 **11.** 0,724 **13.** $3/17$ **15. a)** $1/3$ **b)** $p(M = j \mid W = k) = 1$ se i, j e k forem distintos; $p(M = j \mid W = k) = 0$ se $j = k$ ou $j = i$; $p(M = j \mid W = k) = \frac{1}{2}$ se $i = k$ e $j \neq i$ **c)** $2/3$ **d)** Você deveria trocar de portas, porque agora você tem $2/3$ de chance de ganhar trocando. **17.** A definição de probabilidade condicional nos diz que $p(F_j \mid E) = p(E \cap F_j)/p(E)$. Para o numerador, novamente usando a definição de probabilidade condicional, temos $p(E \cap F_j) = p(E \mid F_j)\, p(F_j)$, como desejado. Para o denominador, mostramos que $p(E) = \sum_{i=1}^{n} p(E \mid F_i) p(F_i)$. Os eventos $E \cap F_i$ são uma partição do evento E; isto é, $(E \cap F_i) \cap (E \cap F_{i_2}) = \emptyset$ quando $i_1 \neq i_2$ (pois os F_i, s são mutuamente excludentes), e $\bigcup_{i=1}^{n} (E \cap F_i) = E$ (pois $\bigcup_{i=1}^{n} F_i = S$). Portanto, $p(E) = \sum_{i=1}^{n} p(E \cap F_i) = \sum_{i=1}^{n} p(E \mid F_i) p(F_i)$. **19.** Não. **21.** Sim. **23.** Pelo Teorema de Bayes, $p(S \mid E_1 \cap E_2) = p(E_1 \cap E_2 \mid S)\, p(S) / [p(E_1 \cap E_2 \mid S)\, p(S) + p(E_1 \cap E_2 \mid \overline{S})\, p(\overline{S})]$. Como não estamos supondo nenhum conhecimento prévio sobre o fato de a mensagem ser ou não um *spam*, fazemos $p(S) = p(\overline{S}) = 0{,}5$, e assim as equações acima se simplificam para $p(S \mid E_1 \cap E_2) = p(E_1 \cap E_2 \mid S) / [p(E_1 \cap E_2 \mid S) + p(E_1 \cap E_2 \mid \overline{S})]$. Por causa da hipótese de independência de E_1, E_2 e S, temos $p(E_1 \cap E_2 \mid S) = p(E_1 \mid S) \cdot p(E_2 \mid S)$, e analogamente para $\overline{S}$.

Seção 6.4

1. 2,5 **3.** $5/3$ **5.** $336/49$ **7.** 170 **9.** $(4n+6)/3$ **11.** $50.700.551/10.077.696 \approx 5{,}03$ **13.** 6 **15.** $p(X \geq j) = \sum_{k=j}^{\infty} p(X = k) = \sum_{k=j}^{\infty} (1-p)^{k-1} p = p(1-p)^{j-1} \sum_{k=0}^{\infty} (1-p)^k = p(1-p)^{j-1}/(1-(1-p)) = (1-p)^{j-1}$ **17.** 2302 **19.** $(7/2) \cdot 7 \neq 329/12$ **21.** $p + (n-1)p(1-p)$ **23.** $5/2$ **25. a)** 0 **b)** n **27. a)** Nos foi dado que X_1 e X_2 são independentes. Para ver que X_1 e X_3 são independentes, enumeramos as oito possibilidades para (X_1, X_2, X_3) e encontramos que (0, 0, 0), (1, 0, 1), (0, 1, 1), (1, 1, 0) têm cada um probabilidade $1/4$ e os outros têm probabilidade 0 (por causa da definição de X_3). Assim, $p(X_1 = 0 \wedge X_3 = 0) = 1/4$, $p(X_1 = 0) = 1/2$ e $p(X_3 = 0) = 1/2$, de modo que é verdade que $p(X_1 = 0 \wedge X_3 = 0) = p(X_1 = 0)\, p(X_3 = 0)$. Essencialmente os mesmos cálculos mostram que $p(X_1 = 0 \wedge X_3 = 1) = p(X_1 = 0)\, p(X_3 = 1)$, $p(X_1 = 1 \wedge X_3 = 0) = p(X_1 = 1)\, p(X_3 = 0)$ e $p(X_1 = 1 \wedge X_3 = 1) = p(X_1 = 1)\, p(X_3 = 1)$. Portanto, por definição, X_1 e X_3 são independentes. O mesmo argumento mostra que X_2 e X_3 são independentes. Para ver que X_3 e $X_1 + X_2$ não são independentes, observamos que $p(X_3 = 1 \wedge X_1 + X_2 = 2) = 0$. Mas $p(X_3 = 1)\, p(X_1 + X_2 = 2) = (1/2)(1/4) = 1/8$. **b)** Vemos dos cálculos na parte (a) que X_1, X_2 e X_3 são todas variáveis aleatórias de Bernoulli, de modo que a variância de cada uma é $(1/2)(1/2) = 1/4$. Portanto, $V(X_1) + V(X_2) + V(X_3) = 3/4$. Usamos os cálculos da parte (a) para ver que $E(X_1 + X_2 + X_3) = 3/2$, e então $V(X_1 + X_2 + X_3) = 3/4$. **c)** Para usar a primeira parte do Teorema 7 para mostrar que $V((X_1 + X_2 + \cdots + X_k) + X_{k+1}) = V(X_1 + X_2 + \cdots + X_k) + V(X_{k+1})$ no passo de indução de uma demonstração por indução matemática, teríamos que saber que $X_1 + X_2 + \cdots + X_k$ e X_{k+1} são independentes, mas vemos da parte (a) que isto não é necessariamente verda-

de. **29.** $1/100$ **31.** $E(X)/a = \sum_r (r/a)\cdot p(X = r) \geq \sum_{r\geq a} 1\cdot p(X = r) = p(X \geq a)$ **33. a)** $10/11$ **b)** 0,9984 **35. a)** Cada uma das $n!$ permutações ocorre com probabilidade $1/n!$, de modo que $E(X)$ é a média sobre todas estas permutações do número de comparações. **b)** Mesmo que o algoritmo continue por $n-1$ rodadas, X será no máximo $n(n-1)/2$. Segue da fórmula para o valor esperado que $E(X) \leq n(n-1)/2$. **c)** O algoritmo prossegue pela comparação de elementos adjacentes e então trocando-os, se necessário. Logo, a única maneira pela qual elementos invertidos podem se tornar não invertidos é eles serem comparados e então trocados. **d)** Como $X(P) \geq I(P)$ para todo P, segue da definição de valor esperado que $E(X) \geq E(I)$. **e)** Esta soma conta 1 para toda ocorrência de inversão. **f)** Isto segue do Teorema 3. **g)** Pelo Teorema 2 com $n = 1$, o valor esperado de $I_{j,k}$ é a probabilidade de que a_k preceda a_j na permutação. Isto é claramente $1/2$ por simetria. **h)** A soma na parte (f) consiste em $C(n, 2) = n(n-1)/2$ termos, cada um dos quais igual a $1/2$, de modo que a soma é $n(n-1)/4$. **i)** Da parte (a) e da parte (b) sabemos que $E(X)$, o objeto de interesse, é no máximo $n(n-1)/2$, e da parte (d) e da parte (h) sabemos que $E(X)$ é no mínimo $n(n-1)/4$, ambos dos quais são $\Theta(n^2)$. **37.** 1 **39.** $V(X+Y) = E((X+Y)^2) - E(X+Y)^2 = E(X^2 + 2XY + Y^2) - [E(X) + E(Y)]^2 = E(X^2) + 2E(XY) + E(Y^2) - E(X)^2 - 2E(X)E(Y) - E(Y)^2 = E(X^2) - E(X)^2 + 2[E(XY) - E(X)E(Y)] + E(Y^2) - E(Y)^2 = V(X) + 2\mathrm{Cov}(X, Y) + V(Y)$ **41.** $[(n-1)/n]^m$ **43.** $(n-1)^m/n^{m-1}$

Exercícios Complementares

1. $1/109.668$ **3. a)** $1/C(52, 13)$ **b)** $4/C(52, 13)$ **c)** $2.944.656/C(52, 13)$ **d)** $35.335.872/C(52, 13)$ **5. a)** 9/2 **b)** 21/4 **7. a)** 9 **b)** 21/2 **9. a)** 8 **b)** 49/6 **11. a)** $n/2^{n-1}$ **b)** $p(1-p)^{k-1}$, em que $p = n/2^{n-1}$ **c)** $2^{n-1}/n$ **13.** $\frac{(m-1)(n-1)+\mathrm{mdc}(m,n)-1}{mn-1}$ **15. a)** 2/3 **b)** 2/3 **17.** 1/32 **19. a)** A probabilidade de alguém ganhar 2^n reais é $1/2^n$, pois isso ocorre precisamente quando o jogador tira $n-1$ coroas seguidas por uma cara. O valor esperado dos ganhos é, portanto, 2^n vezes $1/2^n$ quando n vai 1 para infinito. Como cada um destes termos é 1, a soma é infinita. Em outras palavras, a pessoa deve estar disposta a apostar qualquer quantia de dinheiro e esperar ganhar a longo prazo. **b)** R$ 9, R$ 9 **21. a)** 1/3 quando $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ e $B = \{1, 2, 3, 4\}$; 1/12 quando $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, $A = \{4, 5, 6, 7, 8, 9, 10, 11, 12\}$ e $B = \{1, 2, 3, 4\}$ **b)** 1 quando $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, $A = \{4, 5, 6, 7, 8, 9, 10, 11, 12\}$ e $B = \{1, 2, 3, 4\}$; 3/4 quando $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ e $B = \{1, 2, 3, 4\}$ **23. a)** $p(E_1 \cap E_2) = p(E_1)p(E_2), p(E_1 \cap E_3) = p(E_1)p(E_3), p(E_2 \cap E_3) = p(E_2)\ p(E_3),\ p(E_1 \cap E_2 \cap E_3) = p(E_1)\ p(E_2)\ p(E_3)$ **b)** Sim. **c)** Sim; sim. **d)** Sim; não. **e)** $2^n - n - 1$ **25.** $V(aX + b) = E((aX + b)^2) - E(aX + b)^2 = E(a^2X^2 + 2abX + b^2) - [aE(X) + b]^2 = E(a^2X^2) + E(2abX) + E(b^2) - [a^2E(X)^2 + 2abE(X) + b^2] = a^2E(X^2) + 2abE(X) + b^2 - a^2E(X)^2 - 2abE(X) - b^2 = a^2[E(X^2) - E(X)^2] = a^2V(X)$ **27.** Para contar todo elemento do espaço amostral exatamente uma vez, devemos incluir todo elemento em cada um dos conjuntos e então eliminar a contagem dupla dos elementos nas intersecções. Logo, $p(E_1 \cup E_2 \cup \cdots \cup E_m) = p(E_1) + p(E_2) + \cdots + p(E_m) - p(E_1 \cap E_2) - p(E_1 \cap E_3) - \cdots - p(E_1 \cap E_m) - p(E_2 \cap E_3) - p(E_2 \cap E_4) - \cdots - p(E_2 \cap E_m) - \cdots - p(E_{m-1} \cap E_m) = qm - (m(m-1)/2)r$, pois estão sendo subtraídos $C(m, 2)$ termos. Mas $p(E_1 \cup E_2 \cup \cdots \cup E_m) = 1$, de modo que temos $qm - [m(m-1)/2]r = 1$. Como $r \geq 0$, esta equação nos diz que $qm \geq 1$, de modo que $q \geq 1/m$. Como $q \leq 1$, esta equação também implica que $[m(m-1)/2]r = qm - 1 \leq m - 1$, de onde segue que $r \leq 2/m$. **29. a)** Compramos cartas até termos conseguido uma de cada tipo. Isto significa que compramos X cartas no total. Por outro lado, isto também significa que compramos X_0 cartas até tirarmos o primeiro tipo que conseguimos, e então compramos mais X_1 cartas até tirarmos o segundo tipo que conseguimos, e assim por diante. Logo, X é a soma dos X_j's. **b)** Uma vez que j tipos distintos tenham sido obtidos, existem $n-j$ novos tipos disponíveis de um total de n tipos disponíveis. Como é igualmente provável obtermos cada tipo, a probabilidade de sucesso na próxima compra (tirar um novo tipo) é $(n-j)/n$. **c)** Isto segue imediatamente da definição de distribuição geométrica, da definição de X_j e da parte (b). **d)** Da parte (c) segue que $E(X_j) = n/(n-j)$. Assim, pela linearidade do valor esperado e pela parte (a), temos $E(X) = E(X_0) + E(X_1) + \cdots + E(X_{n-1}) = \frac{n}{n} + \frac{n}{n-1} + \cdots + \frac{n}{1} = n(\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{1})$. **e)** Cerca de 224,46 **31.** $24 \cdot 13^4/(52 \cdot 51 \cdot 50 \cdot 49)$

CAPÍTULO 7

Seção 7.1

1. a) 2, 12, 72, 432, 2592 **b)** 2, 4, 16, 256, 65, 536 **c)** 1, 2, 5, 11, 26 **d)** 1, 1, 6, 27, 204 **e)** 1, 2, 0, 1, 3 **3. a)** 6, 17, 49, 143, 421 **b)** $49 = 5 \cdot 17 - 6 \cdot 6$, $143 = 5 \cdot 49 - 6 \cdot 17$, $421 = 5 \cdot 143 - 6 \cdot 49$ **c)** $5a_{n-1} - 6a_{n-2} = 5(2^{n-1} + 5 \cdot 3^{n-1}) - 6(2^{n-2} + 5 \cdot 3^{n-2}) = 2^{n-2}(10 - 6) + 3^{n-2}(75 - 30) = 2^{n-2} \cdot 4 + 3^{n-2} \cdot 9 \cdot 5 = 2^n + 3^n \cdot 5 = a_n$ **5. a)** Sim **b)** Não **c)** Não **d)** Sim **e)** Sim **f)** Sim **g)** Não **h)** Não **7. a)** $a_{n-1} + 2a_{n-2} + 2n - 9 = -(n-1) + 2 + 2[-(n-2) + 2] + 2n - 9 = -n + 2 = a_n$ **b)** $a_{n-1} + 2a_{n-2} + 2n - 9 = 5(-1)^{n-1} - (n-1) + 2 + 2[5(-1)^{n-2} - (n-2) + 2] + 2n - 9 = 5(-1)^{n-2}(-1 + 2) - n + 2 = a_n$ **c)** $a_{n-1} + 2a_{n-2} + 2n - 9 = 3(-1)^{n-1} + 2^{n-1} - (n-1) + 2 + 2[3(-1)^{n-2} + 2^{n-2} - (n-2) + 2] + 2n - 9 = 3(-1)^{n-2}(-1 + 2) + 2^{n-2}(2 + 2) - n + 2 = a_n$ **d)** $a_{n-1} + 2a_{n-2} + 2n - 9 = 7 \cdot 2^{n-1} - (n-1) + 2 + 2[7 \cdot 2^{n-2} - (n-2) + 2] + 2n - 9 = 2^{n-2}(7 \cdot 2 + 2 \cdot 7) - n + 2 = a_n$ **9. a)** $a_n = 2 \cdot 3^n$ **b)** $a_n = 2n + 3$ **c)** $a_n = 1 + n(n+1)/2$ **d)** $a_n = n^2 + 4n + 4$ **e)** $a_n = 1$ **f)** $a_n = (3^{n+1} - 1)/2$ **g)** $a_n = 5n!$ **h)** $a_n = 2^n n!$ **11. a)** $a_n = 3a_{n-1}$ **b)** 5.904.900 **13. a)** $a_n = n + a_{n-1}$, $a_0 = 0$ **b)** $a_{12} = 78$ **c)** $a_n = n(n+1)/2$ **15.** $B(k) = [1 + (0{,}07/12)]B(k-1) - 100$, com $B(0) = 5000$ **17.** Seja $P(n)$ a afirmação "$H_n = 2^n - 1$." *Passo base:* $P(1)$ é verdadeira, pois $H_1 = 1$. *Passo de indução:* Supo-

nha que $H_n = 2^n - 1$. Então, como $H_{n+1} = 2H_n + 1$, segue que $H_{n+1} = 2(2^n - 1) + 1 = 2^{n+1} - 1$. **19. a)** $a_n = 2a_{n-1} + a_{n-5}$ para $n \geq 5$ **b)** $a_0 = 1$, $a_1 = 2$, $a_3 = 8$, $a_4 = 16$ **c)** 1217 **21.** 9494 **23. a)** $a_n = a_{n-1} + a_{n-2} + 2^{n-2}$ para $n \geq 2$ **b)** $a_0 = 0$, $a_1 = 0$ **c)** 94 **25. a)** $a_n = a_{n-1} + a_{n-2} + a_{n-3}$ para $n \geq 3$ **b)** $a_0 = 1$, $a_1 = 2$, $a_2 = 4$ **c)** 81 **27. a)** $a_n = a_{n-1} + a_{n-2}$ para $n \geq 2$ **b)** $a_0 = 1$, $a_1 = 1$ **c)** 34 **29. a)** $a_n = 2a_{n-1} + 2a_{n-2}$ para $n \geq 2$ **b)** $a_0 = 1$, $a_1 = 3$ **c)** 448 **31. a)** $a_n = 2a_{n-1} + a_{n-2}$ para $n \geq 2$ **b)** $a_0 = 1$, $a_1 = 3$ **c)** 239 **33. a)** $a_n = 2a_{n-1}$ para $n \geq 2$ **b)** $a_1 = 3$ **c)** 96 **35. a)** $a_n = a_{n-1} + a_{n-2}$ para $n \geq 2$ **b)** $a_0 = 1$, $a_1 = 1$ **c)** 89 **37. a)** $R_n = n + R_{n-1}$, $R_0 = 1$ **b)** $R_n = n(n+1)/2 + 1$ **39. a)** $S_n = S_{n-1} + (n^2 - n + 2)/2$, $S_0 = 1$ **b)** $S_n = (n^3 + 5n + 6)/6$ **41.** 64 **43. a)** $a_n = 2a_{n-1} + 2a_{n-2}$ **b)** $a_0 = 1$, $a_1 = 3$ **c)** 1224 **45.** Claramente, $S(m, 1) = 1$ para $m \geq 1$. Se $m \geq n$, então uma função que não seja sobrejetiva do conjunto com m elementos para o conjunto com n elementos pode ser especificada escolhendo o tamanho da imagem, que é um inteiro entre 1 e $n - 1$ inclusive, escolhendo os elementos da imagem, o que pode ser feito de $C(n, k)$ maneiras, e escolhendo uma função sobrejetiva na imagem, o que pode ser feito de $S(m, k)$ maneiras. Portanto, existem $\sum_{k=1}^{n-1} C(n,k)S(m,k)$ funções que não são sobrejetivas. Contudo, existem n^m funções no total, de modo que $S(m, n) = n^m - \sum_{k=1}^{n-1} C(n,k)S(m,k)$. **47. a)** $C_5 = C_0C_4 + C_1C_3 + C_2C_2 + C_3C_1 + C_4C_0 = 1 \cdot 14 + 1 \cdot 5 + 2 \cdot 2 + 5 \cdot 1 + 14 \cdot 1 = 42$ **b)** $C(10, 5)/6 = 42$ **49.** $J(1) = 1$, $J(2) = 1$, $J(3) = 3$, $J(4) = 1$, $J(5) = 3$, $J(6) = 5$, $J(7) = 7$, $J(8) = 1$, $J(9) = 3$, $J(10) = 5$, $J(11) = 7$, $J(12) = 9$, $J(13) = 11$, $J(14) = 13$, $J(15) = 15$, $J(16) = 1$ **51.** Primeiro, suponha que o número de pessoas seja par, digamos $2n$. Depois de dar a volta no círculo uma vez e retornar para a primeira pessoa, como as pessoas nas posições com números pares foram eliminadas, restam exatamente n pessoas e a pessoa atualmente na posição i é a pessoa que estava originalmente na posição $2i - 1$. Portanto, o sobrevivente [originalmente na posição $J(2n)$] está agora na posição $J(n)$; esta era a pessoa que estava na posição $2J(n) - 1$. Portanto, $J(2n) = 2J(n) - 1$. Analogamente, quando existe um número ímpar de pessoas, digamos $2n + 1$, então depois de dar uma volta no círculo uma vez e então eliminar a pessoa 1, restam n pessoas e a pessoa atualmente na posição i é a pessoa que estava na posição $2i + 1$. Portanto, o sobrevivente será o jogador que ocupa atualmente a posição $J(n)$, ou seja, a pessoa que estava originalmente na posição $2J(n) + 1$. Assim, $J(2n + 1) = 2J(n) + 1$. O caso base é $J(1) = 1$. **53.** 73.977.3617 **55.** Estes nove movimentos resolvem o quebra-cabeça: mova o disco 1 do pino 1 para o pino 2; mova o disco 2 do pino 1 para o pino 3; mova o disco 1 do pino 2 para o pino 3; mova o disco 3 do pino 1 para o pino 2; mova o disco 4 do pino 1 para o pino 4; mova o disco 3 do pino 2 para o pino 4; mova o disco 1 do pino 3 para o pino 2; mova o disco 2 do pino 3 para o pino 4; mova o disco 1 do pino 2 para o pino 4. Para ver que pelo menos nove movimentos são necessários, observe primeiro que pelo menos sete movimentos são necessários, não importando quantos pinos estejam presentes: três para desempilhar os discos, um para mover o maior disco 4 e três outros movimentos para empilhá-los novamente. São necessários pelo menos mais dois movimentos, pois, para mover o disco 4 do pino 1 para o pino 4, os outros três discos precisam estar nos pinos 2 e 3, de modo que pelo menos um movimento é necessário para empilhá-los de novo e um movimento para desempilhá-los. **57.** O caso base é obvio. Se $n > 1$, o algoritmo consiste em três estágios. No primeiro estágio, pela hipótese de indução, são usados $R(n - k)$ movimentos para transferir os $n - k$ menores discos para o pino 2. Então, utilizando o algoritmo usual da Torre de Hanói para três pinos, precisamos $2^k - 1$ movimentos para transferir o resto dos discos (os k maiores discos) para o pino 4, evitando o pino 2. Então, novamente pela hipótese de indução, precisamos de $R(n - k)$ movimentos para transferir os $n - k$ menores discos para o pino 4; todos os pinos estão disponíveis para isso, pois os maiores discos, agora no pino 4, não interferem. Isto demonstra a relação de recorrência. **59.** Primeiro observe que $R(n) = \sum_{j=1}^{n} [R(j) - R(j-1)]$ [o que segue do fato de a soma ser telescópica e $R(0) = 0$]. Pelo Exercício 58, essa é a soma de 2^{k-1} para este intervalo de valores de j. Portanto, a soma é $\sum_{i=1}^{k} i2^{i-1}$, exceto que se n não for um número triangular, então os últimos poucos valores quando $i = k$ estão faltando, e isto é o que o último termo na expressão dada representa. **61.** Pelo Exercício 59, $R(n)$ não é maior que $\sum_{i=1}^{k} i2^{i-1}$. Pode ser mostrado que esta soma é igual a $(k+1)2^k - 2^{k+1} + 1$, de modo que ela não é maior que $(k+1)2^k$. Como $n > k(k-1)/2$, a fórmula quadrática pode ser usada para mostrar que $k < 1 + \sqrt{2n}$ para todo $n > 1$. Portanto, $R(n)$ é limitado superiormente por $(1 + \sqrt{2n} + 1)2^{1+\sqrt{2n}} < 8\sqrt{n2}^{\sqrt{2n}}$ para todo $n > 2$. Desse modo, $R(n)$ é $O(\sqrt{n2}^{\sqrt{2n}})$. **63. a)** 0 **b)** 0 **c)** 2 **d)** $2^{n-1} - 2^{n-2}$ **65.** $a_n - 2\nabla a_n + \nabla^2 a_n = a_n - 2(a_n - a_{n-1}) + (\nabla a_n - \nabla a_{n-1}) = -a_n + 2a_{n-1} + [(a_n - a_{n-1}) - (a_{n-1} - a_{n-2})] = -a_n + 2a_{n-1} + (a_n - 2a_{n-1} + a_{n-2}) = a_{n-2}$ **67.** $a_n = a_{n-1} + a_{n-2} = (a_n - \nabla a_n) + (a_n - 2\nabla a_n + \nabla^2 a_n) = 2a_n - 3\nabla a_n + \nabla^2 a_n$, ou $a_n = 3\nabla a_n - \nabla^2 a_n$

Seção 7.2

1. a) Grau3 **b)** Não. **c)** Grau4 **d)** Não. **e)** Não. **f)** Grau 2 **g)** Não. **3. a)** $a_n = 3 \cdot 2^n$ **b)** $a_n = 2$ **c)** $a_n = 3 \cdot 2^n - 2 \cdot 3^n$ **d)** $a_n = 6 \cdot 2^n - 2 \cdot n2^n$ **e)** $a_n = n(-2)^{n-1}$ **f)** $a_n = 2^n - (-2)^n$ **g)** $a_n = (1/2)^{n+1} - (-1/2)^{n+1}$ **5.** $a_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+1}$ **7.** $[2^{n+1} + (-1)^n]/3$ **9. a)** $P_n = 1{,}2P_{n-1} + 0{,}45P_{n-2}$, $P_0 = 100.000$, $P_1 = 120.000$ **b)** $P_n = (250.000/3)(3/2)^n + (50.000/3)(-3/10)^n$ **11. a)** *Passo base:* Para $n = 1$, temos $1 = 0 + 1$, e, para $n = 2$, temos $3 = 1 + 2$. *Passo de indução:* Suponha que seja verdade para $k \leq n$. Então, $L_{n+1} = L_n + L_{n-1} = f_{n-1} + f_{n+1} + f_{n-2} + f_n = (f_{n-1} + f_{n-2}) + (f_{n+1} + f_n) = f_n + f_{n+2}$. **b)** $L_n = \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n$ **13.** $a_n = 8(-1)^n - 3(-2)^n + 4 \cdot 3^n$ **15.** $a_n = 2^5 + 3(-2)^n - 3^n$ **17.** Seja $a_n = C(n, 0) + C(n-1, 1) + \cdots + C(n-k, k)$, em que $k = \lfloor n/2 \rfloor$. Primeiro, suponha que n seja par, de modo que $k = n/2$, e o último termo é $C(k, k)$. Pela Identidade de Pascal, temos $a_n = 1 + C(n-2, 0) + C(n-2, 1) + C(n-3, 1) + C(n-3, 2) + \cdots + C(n-k, k-2) + C(n-k, k-1) + 1 = 1 + C(n-2, 1) + C(n-3, 2) + \cdots + C(n-k, k-1) + C(n-2, 0) + C(n-3, 1) + \cdots + C(n-k, k$

$-2)+1=a_{n-1}+a_{n-2}$, pois $\lfloor (n-1)/2 \rfloor = k-1 = \lfloor (n-2)/2 \rfloor$. Um cálculo semelhante funciona quando n for ímpar. Portanto, $\{a_n\}$ satisfaz a relação de recorrência $a_n = a_{n-1}+a_{n-2}$ para todo inteiro positivo n, $n \geq 2$. Além disso, $a_1 = C(1,0) = 1$ e $a_2 = C(2,0)+C(1,1) = 2$, que são f_2 e f_3. Segue que $a_n = f_{n+1}$ para todo inteiro positivo n. **19.** $a_n = (n^2+3n+5)(-1)^n$ **21.** $(a_{1,0}+a_{1,1}n+a_{1,2}n^2+a_{1,3}n^3)+(a_{2,0}+a_{2,1}n+a_{2,2}n^2)(-2)^n+(a_{3,0}+a_{3,1}n)3^n+a_{4,0}(-4)^n$ **23. a)** $3a_{n-1}+2^n = 3(-2)^n+2^n = 2^n(-3+1) = -2^{n+1} = a_n$ **b)** $a_n = \alpha 3^n - 2^{n+1}$ **c)** $a_n = 3^{n+1}-2^{n+1}$ **25. a)** $A=-1, B=-7$ **b)** $a_n = \alpha 2^n - n - 7$ **c)** $a_n = 11 \cdot 2^n - n - 7$ **27. a)** $p_3n^3+p_2n^2+p_1n+p_0$ **b)** $n^2p_0(-2)^n$ **c)** $n^2(p_1n+p_0)2^n$ **d)** $(p_2n^2+p_1n+p_0)4^n$ **e)** $n^2(p_2n^2+p_1n+p_0)(-2)^n$ **f)** $n^2(p_4n^4+p_3n^3+p_2n^2+p_1n+p_0)2^n$ **g)** p_0 **29. a)** $a_n = \alpha 2^n + 3^{n+1}$ **b)** $a_n = -2 \cdot 2^n + 3^{n+1}$ **31.** $a_n = \alpha 2^n + \beta 3^n - n \cdot 2^{n+1} + 3n/2 + 21/4$ **33.** $a_n = (\alpha + \beta n + n^2 + n^3/6)2^n$ **35.** $a_n = -4 \cdot 2^n - n^2/4 - 5n/2 + 1/8 + (39/8)3^n$ **37.** $a_n = n(n+1)(n+2)/6$ **39. a)** $1, -1, i, -i$ **b)** $a_n = \frac{1}{4} - \frac{1}{4}(-1)^n + \frac{2+i}{4}i^n + \frac{2-i}{4}(-i)^n$ **41. a)** Usando a fórmula para f_n, vemos que $\left| f_n - \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n \right| = \left| \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n \right| < 1/\sqrt{5} < 1/2$. Isto significa que f_n é o inteiro mais próximo de $\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$. **b)** Menor quando n for par; maior quando n for ímpar **43.** $a_n = f_{n-1}+2f_n-1$ **45. a)** $a_n = 3a_{n-1}+4a_{n-2}, a_0 = 2, a_1 = 6$ **b)** $a_n = [4^{n+1}+(-1)^n]/5$ **47. a)** $a_n = 2a_{n+1} + (n-1)10.000$ **b)** $a_n = 70.000 \cdot 2^{n-1} - 10.000n - 10.000$ **49.** $a_n = 5n^2/12 + 13n/12 + 1$ **51.** Veja Capítulo 11, Seção 5, em [Ma93]. **53.** $6^n \cdot 4^{n-1}/n$

Seção 7.3

1. 14 **3.** O primeiro passo é $(1110)_2(1010)_2 = (2^4+2^2)(11)_2\,(10)_2 + 2^2[(11)_2 - (10)_2][(10)_2 - (10)_2] + (2^2+1)(10)_2 \cdot (10)_2$. O produto é $(10001100)_2$. **5.** $C = 50,665C + 729 = 33.979$ **7. a)** 2 **b)** 4 **c)** 7 **9. a)** 79 **b)** 48.829 **c)** 30.517.579 **11.** $O(\log n)$ **13.** $O(n^{\log_3 2})$ **15.** 5 **17. a)** *Passo base:* Se a seqüência tiver apenas um elemento, então a única pessoa na lista é a vencedora. *Passo recursivo:* Divida a lista em duas partes — a primeira metade e a segunda metade — tão igualmente quanto possível. Aplique o algoritmo recursivamente a cada metade para acabar com, no máximo, dois nomes. Então, percorra a lista toda para contar o número de ocorrências de cada um destes nomes para decidir qual, se um deles for, é o vencedor. **b)** $O(n \log n)$ **19. a)** $f(n) = f(n/2)+2$ **b)** $O(\log n)$ **21. a)** 7 **b)** $O(\log n)$ **23. a)**

```
procedure maior soma(a_1, ..., a_n)
  melhor := 0 {subseqüência vazia tem soma 0}
  for i := 1 to n
  begin
    soma := 0
    for j := i + 1 to n
    begin
      soma := soma + a_j
      if soma > melhor then melhor := soma
    end
  end
```

{*melhor* é a maior soma possível de números na lista}

b) $O(n^2)$ **c)** Dividimos a lista na primeira metade e na segunda metade e aplicamos o algoritmo recursivamente para encontrar a maior soma de termos consecutivos em cada metade. A maior soma de termos consecutivos na seqüência toda ou é um destes dois números ou é a soma de uma seqüência de termos consecutivos que atravessa a metade da lista. Para encontrar a maior soma possível de uma seqüência de termos consecutivos que atravessam a metade da lista, começamos no meio e nos movemos para frente para encontrar a maior soma possível na segunda metade da lista, e nos movemos para trás para encontrar a maior soma possível na primeira metade da lista; a soma procurada é a soma destas duas quantidades. A resposta final é então a maior entre esta soma e as duas respostas obtidas recursivamente. O caso base é que a maior soma de uma seqüência de um termo é a maior entre este número e 0. **d)** 11, 9, 14 **e)** $S(n) = 2S(n/2)+n$, $C(n) = 2C(n/2)+n+2$, $S(1) = 0$, $C(1) = 1$ **f)** $O(n \log n)$, melhor que $O(n^2)$ **25.** (1, 6) e (3, 6) à distância 2 **27.** O algoritmo é essencialmente o mesmo que o algoritmo dado no Exemplo 12. A faixa central ainda tem largura $2d$, mas precisamos considerar apenas duas caixas de tamanho $d \times d$, em vez de oito caixas de tamanho $(d/2) \times (d/2)$. A relação de recorrência é a mesma do Exemplo 12, exceto que o coeficiente 7 é substituído por 1. **29.** Com $k = \log_b n$, segue que $f(n) = a^k f(1) + \sum_{j=0}^{k-1} a^j c(n/b^j)^d = a^k f(1) + \sum_{j=0}^{k-1} cn^d = a^k f(1) + kcn^d = a^{\log_b n} f(1) + c(\log_b n)n^d = n^{\log_b a} f(1) + cn^d \log_b n = n^d f(1) + cn^d \log_b n$. **31.** Seja $k = \log_b n$, em que n é uma potência de b. *Passo base:* Se $n = 1$ e $k = 0$, então $c_1 n^d + c_2 n^{\log_b a} = c_1 + c_2 = b^d c/(b^d - a) + f(1) + b^d c/(a - b^d) = f(1)$. *Passo de indução:* Suponha que seja verdadeira para k, em que $n = b^k$. Então, para $n = b^{k+1}$, $f(n) = af(n/b) + cn^d = a\{[b^d c/(b^d - a)](n/b)^d + [f(1) + b^d c/(a - b^d)] \cdot (n/b)^{\log_b a}\} + cn^d = b^d c/(b^d - a)n^d a/b^d + [f(1) + b^d c/(a - b^d)]n^{\log_b a} + cn^d = n^d[ac/(b^d - a) + c(b^d - a)/(b^d - a)] + [f(1) + b^d c/(a - b^d c)]n^{\log_b a} = [b^d c/(b^d - a)]n^d + [f(1) + b^d c/(a - b^d)]n^{\log_b a}$. **33.** Se $a > b^d$, então $\log_b a > d$, de modo que o segundo termo domina, fornecendo $O(n^{\log_b a})$. **35.** $O(^{\log_4 5})$ **37.** $O(n^3)$

Seção 7.4

1. $f(x) = 2(x^6-1)/(x-1)$ **3. a)** $f(x) = 2x(1-x^6)/(1-x)$ **b)** $x^3/(1-x)$ **c)** $x/(1-x^3)$ **d)** $2/(1-2x)$ **e)** $(1+x)^7$ **f)** $2/(1+x)$ **g)** $[1/(1-x)] - x^2$ **h)** $x^3/(1-x)^2$ **5. a)** $5/(1-x)$ **b)** $1/(1-3x)$ **c)** $2x^3/(1-x)$ **d)** $(3-x)/(1-x)^2$ **e)** $(1+x)^8$ **f)** $1/(1-x)^5$ **7. a)** $a_0 = -64$, $a_1 = 144$, $a_2 = -108$, $a_3 = 27$ e $a_n = 0$ para todo $n \geq 4$ **b)** Os únicos coeficientes não nulos são $a_0 = 1$, $a_3 = 3$, $a_6 = 3$, $a_9 = 1$. **c)** $a_n = 5^n$ **d)** $a_n = (-3)^{n-3}$ para $n \geq 3$, e $a_0 = a_1 = a_2 = 0$ **e)** $a_0 = 8$, $a_1 = 3$, $a_2 = 2$, $a_n = 0$ para n ímpar maior que 2 e $a_n = 1$ para n par maior que 2 **f)** $a_n = 1$ se n for um múltiplo positivo de 4, $a_n = -1$ se $n < 4$ e $an = 0$ nos outros casos **g)** $a_n = n-1$ para $n \geq 2$ e $a_0 = a_1 = 0$ **h)** $a_n = 2^{n+1}/n!$ **9. a)** 6 **b)** 3 **c)** 9 **d)** 0 **e)** 5 **11. a)** 1024 **b)** 11 **c)** 66 **d)** 292.864 **e)** 20.412 **13.** 10 **15.** 50 **17.** 20 **19.** $f(x) = 1/[(1-x)(1-x^2)(1-x^5)(1-x^{10})]$

21. 15 **23. a)** $x^4(1+x+x^2+x^3)^2/(1-x)$ **b)** 6 **25. a)** O coeficiente de x^r na expansão em série de potências de $1/[(1-x^3)(1-x^4)(1-x^{20})]$ **b)** $1/(1-x^3-x^4-x^{20})$ **c)** 7 **d)** 3224 **27. a)** 3 **b)** 29 **c)** 29 **d)** 242 **29. a)** 10 **b)** 49 **c)** 2 **d)** 4 **31. a)** $G(x)-a_0-a_1x-a_2x^2$ **b)** $G(x^2)$ **c)** $x^4G(x)$ **d)** $G(2x)$ **e)** $\int_0^x G(t)dt$ **f)** $G(x)/(1-x)$ **33.** $a_k=2\cdot 3^k-1$ **35.** $a_k=18\cdot 3^k-12\cdot 2^k$ **37.** $a_k=k^2+8k+20+(6k-18)2^k$ **39.** Seja $G(x)=\sum_{k=0}^{\infty}f_kx^k$. Depois de trocar o índice da somatória e somar as séries, vemos que $G(x)-xG(x)-x^2G(x)=f_0+(f_1-f_0)x+\sum_{k=2}^{\infty}(f_k-f_{k-1}-f_{k-2})x^k=0+x+\sum_{k=2}^{\infty}0x^k$. Portanto, $G(x)-xG(x)-x^2G(x)=x$. Isolando $G(x)$, obtemos $G(x)=x/(1-x-x^2)$. Pelo método das frações parciais, pode ser mostrado que $x/(1-x-x^2)=(1/\sqrt{5})[1/(1-\alpha x)-1/(1-\beta x)]$, em que $\alpha=(1+\sqrt{5})/2$ e $\beta=(1-\sqrt{5})/2$. Usando o fato de que $1/(1-\alpha x)=\sum_{k=0}^{\infty}\alpha^kx^k$, segue que $G(x)=(1/\sqrt{5})\cdot\sum_{k=0}^{\infty}(\alpha^k-\beta^k)x^k$. Portanto, $f_k=(1/\sqrt{5})\cdot(\alpha^k-\beta^k)$. **41. a)** Seja $G(x)=\sum_{n=0}^{\infty}C_nx^n$ a função geradora de $\{C_n\}$. Então, $G(x)^2=\sum_{n=0}^{\infty}(\sum_{k=0}^{n}C_kC_{n-k})x^n=\sum_{n=1}^{\infty}(\sum_{k=0}^{n-1}C_kC_{n-1-k})x^{n-1}=\sum_{n=1}^{\infty}C_nx^{n-1}$. Logo, $xG(x)^2=\sum_{n=1}^{\infty}C_nx^n$, o que implica que $xG(x)^2-G(x)+1=0$. A aplicação da fórmula quadrática mostra que $G(x)=\frac{1\pm\sqrt{1-4x}}{2x}$. Escolhemos o sinal de menos nesta fórmula porque a escolha do sinal de mais leva a uma divisão por zero. **b)** Pelo Exercício 40, $(1-4x)^{-1/2}=\sum_{n=0}^{\infty}\binom{2n}{n}x^n$. Integrar termo a termo (o que é válido por um teorema do cálculo), mostra que $\int_0^x(1-4t)^{-1/2}dt=\sum_{n=0}^{\infty}\binom{2n}{n}x^n\cdot\frac{1}{n+1}\binom{2n}{n}x^{n+1}=x\sum_{n=0}^{\infty}\frac{1}{n+1}\binom{2n}{n}x^n$. Portanto, $\int_0^x(1-4t)^{-1/2}dt=\frac{1-\sqrt{1-4x}}{2}=xG(x)$, ao igualarmos os coeficientes, obtemos $C_n=\frac{1}{n+1}\binom{2n}{n}$. **43.** Aplicando o Teorema Binomial à igualdade $(1+x)^{m+n}=(1+x)^m(1+x)^n$, obtemos $\sum_{r=0}^{m+n}C(m+n,r)x^r=\sum_{r=0}^{m}C(m,r)x^r\cdot\sum_{r=0}^{n}C(n,r)x^r=\sum_{r=0}^{m+n}\left[\sum_{k=0}^{r}C(m,r-k)\cdot C(n,k)\right]x^r$. A comparação dos coeficientes fornece a identidade desejada. **45. a)** $2e^x$ **b)** e^{-x} **c)** e^{3x} **d)** xe^x+e^x **e)** $(e^x-1)/x$ **47. a)** $a_n=(-1)^n$ **b)** $a_n=3\cdot 2^n$ **c)** $a_n=3^n-3\cdot 2^n$ **d)** $a_n=(-2)^n$ para $n\geq 2$, $a_1=-3$, $a_0=2$ **e)** $a_n=(-2)^n+n!$ **f)** $a_n=(-3)^n+n!\cdot 2^n$ para $n\geq 2$, $a_0=1$, $a_1=-2$ **g)** $a_n=0$ se n for ímpar e $a_n=n!/(n/2)!$ se n for par **49. a)** $a_n=6a_{n-1}+8^{n-1}$ para $n\geq 1$, $a_0=1$ **b)** A solução geral da relação de recorrência associada linear e homogênea é $a_n^{(h)}=\alpha 6^n$. Uma solução particular é $a_n^{(p)}=\frac{1}{2}\cdot 8^n$. Portanto, a solução geral é $a_n=\alpha 6^n+\frac{1}{2}\cdot 8^n$. Usando a condição inicial, segue que $\alpha=\frac{1}{2}$. Portanto, $a_n=(6^n+8^n)/2$. **c)** Seja $G(x)=\sum_{k=0}^{\infty}a_kx^k$. Usando a relação de recorrência para $\{a_k\}$, pode ser mostrado que $G(x)-6xG(x)=(1-7x)/(1-8x)$. Portanto, $G(x)=(1-7x)/[(1-6x)(1-8x)]$. Usando frações parciais, segue que $G(x)=(1/2)/(1-6x)+(1/2)/(1-8x)$. Com a ajuda da Tabela 1, segue que $a_n=(6^n+8^n)/2$. **51.** $\frac{1}{1-x}\cdot\frac{1}{1-x^2}\cdot\frac{1}{1-x^3}\cdots$ **53.** $(1+x)(1+x)^2(1+x)^3\cdots$ **55.** As funções geradoras obtidas nos exercícios 52 e 53 são iguais porque $(1+x)(1+x^2)(1+x^3)\cdots=\frac{1-x^2}{1-x}\cdot\frac{1-x^4}{1-x^2}\cdot\frac{1-x^6}{1-x^3}\cdots=\frac{1}{1-x}\cdot\frac{1}{1-x^3}\cdot\frac{1}{1-x^5}\cdots$. **57. a)** $G_X(1)=\sum_{k=0}^{\infty}p(X=k)\cdot 1^k=\sum_{k=0}^{\infty}P(X=k)=1$ **b)** $G'_X(1)=\frac{d}{dx}\sum_{k=0}^{\infty}p(X=k)\cdot x^k|_{x=1}=\sum_{k=0}^{\infty}p(X=k)\cdot k\cdot x^{k-1}|_{x=1}=\sum_{k=0}^{\infty}p(X=k)\cdot k=E(X)$ **c)** $G''_X(1)=\frac{d^2}{dx^2}\sum_{k=0}^{\infty}p(X=k)\cdot x^k|_{x=1}=\sum_{k=0}^{\infty}p(X=k)\cdot k(k-1)\cdot x^{k-2}|_{x=1}=\sum_{k=0}^{\infty}p(X=k)\cdot(k^2-k)=V(X)+E(X)^2-E(X)$. Combinando isto com a parte (b), obtemos os resultados desejados. **59. a)** $G(x)=p^m/(1-qx)^m$ **b)** $V(x)=mq/p^2$

Seção 7.5

1. a) 30 **b)** 29 **c)** 24 **d)** 18 **3.** 1% **5. a)** 300 **b)** 150 **c)** 175 **d)** 100 **7.** 492 **9.** 974 **11.** 55 **13.** 248 **15.** 50.138 **17.** 234 **19.** $|A_1\cup A_2\cup A_3\cup A_4\cup A_5|=|A_1|+|A_2|+|A_3|+|A_4|+|A_5|-|A_1\cap A_2|-|A_1\cap A_3|-|A_1\cap A_4|-|A_1\cap A_5|-|A_2\cap A_3|-|A_2\cap A_4|-|A_2\cap A_5|-|A_3\cap A_4|-|A_3\cap A_5|-|A_4\cap A_5|+|A_1\cap A_2\cap A_3|+|A_1\cap A_2\cap A_4|+|A_1\cap A_2\cap A_5|+|A_1\cap A_3\cap A_4|+|A_1\cap A_3\cap A_5|+|A_1\cap A_4\cap A_5|+|A_2\cap A_3\cap A_4|+|A_2\cap A_3\cap A_5|+|A_2\cap A_4\cap A_5|+|A_3\cap A_4\cap A_5|-|A_1\cap A_2\cap A_3\cap A_4|-|A_1\cap A_2\cap A_3\cap A_5|-|A_1\cap A_2\cap A_4\cap A_5|-|A_1\cap A_3\cap A_4\cap A_5|-|A_2\cap A_3\cap A_4\cap A_5|+|A_1\cap A_2\cap A_3\cap A_4\cap A_5|$ **21.** $|A_1\cup A_2\cup A_3\cup A_4\cup A_5\cup A_6|=|A_1|+|A_2|+|A_3|+|A_4|+|A_5|+|A_6|-|A_1\cap A_2|-|A_1\cap A_3|-|A_1\cap A_4|-|A_1\cap A_5|-|A_1\cap A_6|-|A_2\cap A_3|-|A_2\cap A_4|-|A_2\cap A_5|-|A_2\cap A_6|-|A_3\cap A_4|-|A_3\cap A_5|-|A_3\cap A_6|-|A_4\cap A_5|-|A_4\cap A_6|-|A_5\cap A_6|$ **23.** $p(E_1\cup E_2\cup E_3)=p(E_1)+p(E_2)+p(E_3)-p(E_1\cap E_2)-p(E_1\cap E_3)-p(E_2\cap E_3)+p(E_1\cap E_2\cap E_3)$ **25.** 4972/71.295 **27.** $p(E_1\cup E_2\cup E_3\cup E_4\cup E_5)=p(E_1)+p(E_2)+p(E_3)+p(E_4)+p(E_5)-p(E_1\cap E_2)-p(E_1\cap E_3)-p(E_1\cap E_4)-p(E_1\cap E_5)-p(E_2\cap E_3)-p(E_2\cap E_4)-p(E_2\cap E_5)-p(E_3\cap E_4)-p(E_3\cap E_5)-p(E_4\cap E_5)+p(E_1\cap E_2\cap E_3)+p(E_1\cap E_2\cap E_4)+p(E_1\cap E_2\cap E_5)+p(E_1\cap E_3\cap E_4)+p(E_1\cap E_3\cap E_5)+p(E_1\cap E_4\cap E_5)+p(E_2\cap E_3\cap E_4)+P(E_2\cap E_3\cap E_5)+p(E_2\cap E_4\cap E_5)+p(E_3\cap E_4\cap E_5)$ **29.** $p(\bigcup_{i=1}^{n}E_i)=\sum_{1\leq i\leq n}p(E_i)-\sum_{1\leq i<j\leq n}p(E_i\cap E_j)+\sum_{1\leq i<j<k\leq n}p(E_i\cap E_j\cap E_k)-\cdots+(-1)^{n+1}p(\bigcap_{i=1}^{n}E_i)$

Seção 7.6

1. 75 **3.** 6 **5.** 46 **7.** 9875 **9.** 540 **11.** 2100 **13.** 1854 **15. a)** $D_{100}/100!$ **b)** $100D_{99}/100!$ **c)** $C(100,2)/100!$ **d)** 0 **e)** $1/100!$ **17.** 2.170.680 **19.** Pelo Exercício 18, temos $D_n-nD_{n-1}=-[D_{n-1}-(n-1)D_{n-2}]$. Iterando, temos $D_n-nD_{n-1}=-[D_{n-1}-(n-1)D_{n-2}]=-[-(D_{n-2}-(n-2)D_{n-3})]=D_{n-2}-(n-2)D_{n-3}=\cdots=(-1)^n(D_2-2D_1)=(-1)^n$, pois $D_2=1$ e $D_1=0$. **21.** Quando n é ímpar. **23.** $\frac{\phi(n)}{n}=\frac{p_1p_2\cdots p_m}{n}=1-\sum_{i=1}^{m}\frac{p_i}{p_i}+\sum_{1\leq i<j\leq m}\frac{p_ip_j}{p_ip_j}-\cdots\pm\prod_{i=1}^{m}(1-\frac{1}{p_i})$ **25.** 4 **27.** Existem n^m funções de um conjunto com m elementos em um conjunto com n elementos, $C(n,1)(n-1)^m$ funções de um conjunto com m elementos para um conjunto com n elementos que não atinja exatamente um elemento, $C(n,2)(n-2)^m$ funções de um conjunto com m elementos para um conjunto com n elementos que não atinja exatamente dois elementos, e assim por diante, com $C(n,n-1)\cdot 1^m$ funções de um conjunto com m elementos para um conjunto com n elementos que não atinja exatamente

$n-1$ elementos. Portanto, pelo princípio de exclusão–inclusão, existem $n^m - C(n,1)(n-1)^m + C(n,2)(n-2)^m - \cdots + (-1)^{n-1}C(n,n-1)\cdot 1^m$ funções sobrejetivas.

Exercícios Complementares

1. **a)** $A_n = 4A_{n-1}$ **b)** $A_1 = 40$ **c)** $A_n = 10 \cdot 4^n$ **3.** **a)** $M_n = M_{n-1} + 160.000$ **b)** $M_1 = 186.000$ **c)** $M_n = 160.000n + 26.000$ **d)** $T_n = T_{n-1} + 160.000n + 26.000$ **e)** $T_n = 80.000n^2 + 106.000n$ **5.** **a)** $a_n = a_{n-2} + a_{n-3}$ **b)** $a_1 = 0$, $a_2 = 1$, $a_3 = 1$ **c)** $a_{12} = 12$ **7.** **a)** 2 **b)** 5 **c)** 8 **d)** 16 **9.** $a_n = 2^n$ **11.** $a_n = 2 + 4n/3 + n^2/2 + n^3/6$ **13.** $a_n = a_{n-2} + a_{n-3}$ **15.** $O(n^4)$ **17.** $O(n)$ **19.** **a)** $18n + 18$ **b)** 18 **c)** 0 **21.** $\Delta(a_n b_n) = a_{n+1}b_{n+1} - a_n b_n = a_{n+1}(b_{n+1} - b_n) + b_n(a_{n+1} - a_n) = a_{n+1}\Delta b_n + b_n \Delta a_n$ **23.** **a)** Seja $G(x) = \sum_{n=0}^{\infty} a_n x^n$. Então, $G'(x) = \sum_{n=1}^{\infty} n a_n x^{n-1} = \sum_{n=0}^{\infty} (n+1)a_{n+1}x^n$. Portanto, $G'(x) - G'(x) = \sum_{n=0}^{\infty}[(n+1)a_{n+1} - a_n]x^n = \sum_{n=0} x^n/n! = e^x$, como desejado. Que $G(0) = a_0 = 1$ foi dado. **b)** Temos $[e^{-x}G(x)]' = e^{-x}G'(x) - e^{-x}G(x) = e^{-x}[G'(x) - G(x)] = e^{-x}\cdot e^x = 1$. Portanto, $e^{-x}G(x) = x + c$, em que c é uma constante. Conseqüentemente, $G(x) = xe^x + ce^x$. Como $G(0) = 1$, segue que $c = 1$. **c)** Temos $G(x) = \sum_{n=0}^{\infty} x^{n+1}/n! + \sum_{n=0}^{\infty} x^n/n! = \sum_{n=1}^{\infty} x^n/(n-1)! + \sum_{n=0}^{\infty} x^n/n!$. Portanto, $a_n = 1/(n-1)! + 1/n!$ para todo $n \geq 1$, e $a_0 = 1$. **25.** 7 **27.** 110 **29.** 0 **31.** **a)** 19 **b)** 65 **c)** 122 **d)** 167 **e)** 168 **33.** $D_{n-1}/(n-1)!$ **35.** 11/32

CAPÍTULO 8

Seção 8.1

1. **a)** {(0, 0), (1, 1), (2, 2), (3, 3)} **b)** {(1, 3), (2, 2), (3, 1), (4, 0)} **c)** {(1, 0), (2, 0), (2, 1), (3, 0), (3, 1), (3, 2), (4, 0), (4, 1), (4, 2), (4, 3)} **d)** {(1, 0), (1, 1), (1, 2), (1, 3), (2, 0), (2, 2), (3, 0), (3, 3), (4, 0)} **e)** {(0, 1), (1, 0), (1, 1), (1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2), (4, 1), (4, 3)} **f)** {(1, 2), (2, 1), (2, 2)} **3.** **a)** Transitiva **b)** Reflexiva, simétrica, transitiva **c)** Simétrica **d)** Anti-simétrica **e)** Reflexiva, simétrica, anti-simétrica, transitiva **f)** Nenhuma dessas propriedades **5.** **a)** Reflexiva, transitiva **b)** Simétrica **c)** Simétrica **d)** Simétrica **7.** **a)** Simétrica **b)** Simétrica, transitiva **c)** Simétrica **d)** Reflexiva, simétrica, transitiva **e)** Reflexiva, transitiva **f)** Reflexiva, simétrica, transitiva **g)** Anti-simétrica **h)** Anti-simétrica, transitiva **9.** (c), (d), (f) **11.** **a)** Não é irreflexiva **b)** Não é irreflexiva **c)** Não é irreflexiva **d)** Não é irreflexiva **13.** Sim, por exemplo {(1, 1)} em {1, 2} **15.** $(a, b) \in R$ se e somente se a for mais alto que b. **17.** (a) **19.** Nenhuma. **21.** $\forall a \forall b\,[(a, b) \in R \rightarrow (b, a) \notin R]$ **23.** 2^{mn} **25.** **a)** {(a, b) | b divide a} **b)** {(a, b) | a não divide b} **27.** O gráfico de f^{-1}. **29.** **a)** {(a, b) | a precisa ler ou já leu b} **b)** {(a, b) | a precisa ler ou já leu b} **c)** {(a, b) | ou a precisa ler b, mas ainda não o leu ou a já leu b, mas não precisava lê-lo} **d)** {(a, b) | a precisa ler b, mas ainda não o leu} **e)** {(a, b) | a já leu b, mas não precisava lê-lo} **31.** $S \circ R = \{(a, b) \mid a$ é progenitor de b e b tem um irmão$\}$, $R \circ S = \{(a, b) \mid a$ é tia ou tio de $b\}$ **33.** **a)** $\mathbf{R}^2$ **b)** R_6 **c)** R_3 **d)** R_3 **e)** $\emptyset$ **f)** R_1 **g)** R_4 **h)** R_4 **35.** **a)** R_1 **b)** R_2 **c)** R_3 **d)** $\mathbf{R}^2$ **e)** R_3 **f)** $\mathbf{R}^2$ **g)** $\mathbf{R}^2$ **h)** $\mathbf{R}^2$ **37.** b terminou seu doutorado sob a orientação de alguém que terminou seu doutorado sob a orientação de a; existe uma seqüência de $n+1$ pessoas, começando com a e terminando com b, tal que cada uma é orientadora da próxima pessoa na seqüência **39.** **a)** {(a, b) | $a - b \equiv 0, 3, 4, 6, 8$ ou 9 (mod 12)} **b)** {(a, b) | $a \equiv b$ (mod 12)} **c)** {(a, b) | $a - b \equiv 3, 6$ ou 9 (mod 12)} **d)** {(a, b) | $a - b \equiv 4$ ou 8 (mod 12)} **e)** {(a, b) | $a - b \equiv 3, 4, 6, 8$ ou 9 (mod 12)} **41.** 8 **43.** **a)** 65536 **b)** 32768 **45.** **a)** $2^{n(n+1)/2}$ **b)** $2^n 3^{n(n-1)/2}$ **c)** $3^{n(n-1)/2}$ **d)** $2^{n(n-1)}$ **e)** $2^{n(n-1)/2}$ **f)** $2^{n^2} - 2 \cdot 2^{n(n-1)}$ **47.** Não pode existir tal b. **49.** Se R for simétrica e $(a, b) \in R$, então $(b, a) \in R$, de modo que $(a, b) \in R^{-1}$. Logo, $R \subseteq R^{-1}$. Analogamente, $R^{-1} \subseteq R$. Assim, $R = R^{-1}$. Reciprocamente, se $R = R^{-1}$ e $(a, b) \in R$, então $(a, b) \in R^{-1}$, de modo que $(b, a) \in R$. Portanto, R é simétrica. **51.** R é reflexiva se e somente se $(a, a) \in R$ para todo $a \in A$ se e somente se $(a, a) \in R^{-1}$ [pois $(a, a) \in R$ se e somente se $(a, a) \in R^{-1}$] se e somente se R^{-1} é reflexiva. **53.** Use indução matemática. O resultado é trivial para $n = 1$. Suponha que R^n seja reflexiva e transitiva. Pelo Teorema 1, $R^{n+1} \subseteq R$. Para ver que $R \subseteq R^{n+1} = R^n \circ R$, seja $(a, b) \in R$. Pela hipótese de indução, $R^n = R$ e, logo, é reflexiva. Assim, $(b, b) \in R^n$. Portanto, $(a, b) \in R^{n+1}$. **55.** Use indução matemática. O resultado é trivial para $n = 1$. Suponha que R^n seja reflexiva. Então $(a, a) \in R^n$ para todo $a \in A$ e $(a, a) \in R$. Portanto, $(a, a) \in R^n \circ R = R^{n+1}$ para todo $a \in A$. **57.** Não, por exemplo, considere $R = \{(1, 2), (2, 1)\}$.

Seção 8.2

1. {(1, 2, 3), (1, 2, 4), (1, 3, 4), (2, 3, 4)} **3.** (Nadir, 122, 34, Detroit, 08:10), (Acme, 221, 22, Denver, 08:17), (Acme, 122, 33, Anchorage, 08:22), (Acme, 323, 34, Honolulu, 08:30), (Nadir, 199, 13, Detroit, 08:47), (Acme, 222, 22, Denver, 09:10), (Nadir, 322, 34, Detroit, 09:44) **5.** Companhia aérea e número do vôo, companhia aérea e horário de partida **7.** **a)** Sim. **b)** Não. **c)** Não. **9.** **a)** Número do Social Security **b)** Não existem duas pessoas com o mesmo nome que tenham o mesmo endereço **c)** Não existem duas pessoas com o mesmo nome que morem juntas. **11.** (Nadir, 122, 34, Detroit, 08:10), (Nadir, 199, 13, Detroit, 08:47), (Nadir, 322, 34, Detroit, 09:44) **13.** (Nadir, 122, 34, Detroit, 08:10), (Nadir, 199, 13, Detroit, 08:47), (Nadir, 322, 34, Detroit, 09:44), (Acme, 221, 22, Denver, 08:17), (Acme, 222, 22, Denver, 09:10)

15. $P_{3,5,6}$

17.

Companhia aerea	*Destino*
Nadir	Detroit
Acme	Denver
Acme	Anchorage
Acme	Honolulu

19.

Fornecedor	*Numero da parte*	*Projeto*	*Quantidade*	*Codigo de cor*
23	1092	1	2	2
23	1101	3	1	1
23	9048	4	12	2
31	4975	3	6	2
31	3477	2	25	2
32	6984	4	10	1
32	9191	2	80	4
33	1001	1	14	8

21. Ambos os lados desta equação escolhem o subconjunto de R que consiste naquelas n-uplas que satisfaçam ambas as condições C_1 e C_2. **23.** Ambos os lados desta equação escolhem o conjunto das n-uplas que estão em R, estão em S e satisfazem a condição C. **25.** Ambos os lados desta equação escolhem as m-uplas que consistem nas i_1-ésima, i_2-ésima, …, i_m-*ésima* componentes das n-uplas em R ou S. **27.** Sejam $R = \{(a, b)\}$ e $S = \{(a, c)\}$, $n = 2$, $m = 1$ e $i_1 = 1$; $P_1(R - S) = \{(a)\}$, mas $P_1(R) - P_1(S) = \emptyset$. **29. a)** J_2 seguido por $P_{1,3}$ **b)** (23, 1), (23, 3), (31, 3), (32, 4) **31.** Não existe nenhuma chave primária.

Seção 8.3

1. a) $\begin{bmatrix}1&1&1\\0&0&0\\0&0&0\end{bmatrix}$ **b)** $\begin{bmatrix}0&1&0\\1&1&0\\0&0&1\end{bmatrix}$

c) $\begin{bmatrix}1&1&1\\0&1&1\\0&0&1\end{bmatrix}$ **d)** $\begin{bmatrix}0&0&1\\0&0&0\\1&0&0\end{bmatrix}$

3. a) (1, 1), (1, 3), (2, 2), (3, 1), (3, 3) **b)** (1, 2), (2, 2), (3, 2) **c)** (1, 1), (1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2), (3, 3) **5.** A relação é irreflexiva se e somente se a diagonal principal da matriz contiver apenas 0s. **7. a)** Reflexiva, simétrica, transitiva **b)** Anti-simétrica, transitiva **c)** Simétrica **9. a)** 4950 **b)** 9900 **c)** 99 **d)** 100 **e)** 1 **11.** Mude cada 0 para 1 e cada 1 para 0.

13. a) $\begin{bmatrix}0&1&1\\1&1&0\\1&0&1\end{bmatrix}$ **b)** $\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}$ **c)** $\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$

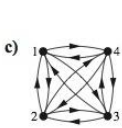
15. a) $\begin{bmatrix}0&0&1\\1&1&0\\0&1&1\end{bmatrix}$ **b)** $\begin{bmatrix}1&1&0\\0&1&1\\1&1&1\end{bmatrix}$ **c)** $\begin{bmatrix}0&1&1\\1&1&1\\1&1&1\end{bmatrix}$

17. $n^2 - k$

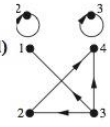
19. a)

b)

c)

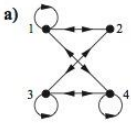


d)

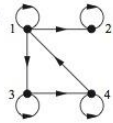


21. Por simplicidade, indicamos pares de arestas entre os mesmos dois vértices em sentidos opostos usando uma flecha dupla, em vez de desenhar duas curvas separadas.

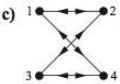
a)



b)



c)

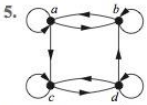


23. $\{(a, b), (a, c), (b, c), (c, b)\}$ **25.** $(a, c), (b, a), (c, d), (d, b)$ **27.** $\{(a, a), (a, b), (a, c), (b, a), (b, b), (b, c), (c, a), (c, b), (d, d)\}$ **29.** A relação é assimétrica se e somente se o grafo orientado não tiver nenhum laço e nenhum caminho fechado de comprimento 2. **31.** Exercício 23: irreflexiva. Exercício 24: reflexiva, anti-simétrica, transitiva. Exercício 25: irreflexiva, anti-simétrica. **33.** Inverta o sentido de todas as arestas do dígrafo para R. **35.** Demonstração por indução matemática. *Passo base:* Trivial para $n = 1$. *Passo de indução:* Suponha que seja verdadeira para k. Como $R^{k+1} = R^k \circ R$, sua matriz é $\mathbf{M}_R \odot \mathbf{M}_{R^k}$. Pela hipótese de indução, isto é $\mathbf{M}_R \odot \mathbf{M}_R^{[k]} = \mathbf{M}_R^{[k+1]}$.

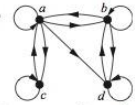
Seção 8.4

1. a) $\{(0, 0), (0, 1), (1, 1), (1, 2), (2, 0), (2, 2), (3, 0), (3, 3)\}$ **b)** $\{(0, 1), (0, 2), (0, 3), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2), (3, 0)\}$ **3.** $\{(a, b) \mid a$ divide b ou b divide $a\}$

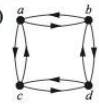
5.



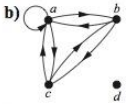
7.



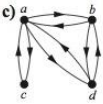
9. a)



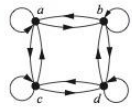
b)



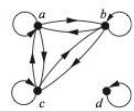
c)



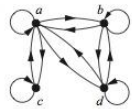
11. a)



b)



c)



13. O fecho simétrico de R é $R \cup R^{-1}$. $\mathbf{M}_{R \cup R^{-1}} = \mathbf{M}_R \vee \mathbf{M}_{R^{-1}} = \mathbf{M}_R \vee \mathbf{M}'_R$. **15.** Apenas quando R é irreflexiva, em cujo caso ela é seu próprio fecho. **17.** *a, a, a, a*; *a, b, e, a*; *a, d, e, a*; *b, c, c, b*; *b, e, a, b*; *c, b, c, c*; *c, c, b, c*; *c, c, c, c*; *d, e, a, d*; *d, e, e, d*; *e, a, b, e*; *e, a, d, e*; *e, d, e, e*; *e, e, d, e*; *e, e, e, e* **19. a)** {(1, 1), (1, 5), (2, 3), (3, 1), (3, 2), (3, 3), (3, 4), (4, 1), (4, 5), (5, 3), (5, 4)} **b)** {(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 5), (3, 1), (3, 3), (3, 4), (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (5, 1), (5, 3), (5, 5)} **c)** {(1, 1), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (4, 1), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)} **d)** {(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)} **e)** {(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)} **f)** {(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (4, 1), (4, 2), (4,3),(4,4),(4,5),(5,1),(5,2),(5,3),(5,4),(5,5)} **21. a)** Se existir um estudante c que tenha uma aula em comum que a e tenha uma aula um comum que b **b)** Se existirem dois estudantes c e d tais que a e c tenham uma aula em comum, c e d tenham uma aula em comum e d e b tenham uma aula em comum **c)** Se existir uma seqüência $s_0, \ldots, s_n$ de estudantes com $n \geq 1$ tais que $s_0 = a$, $s_n = b$, e para cada $i = 1, 2, \ldots, n$, s_i e s_{i-1} tenham uma aula em comum **23.** O resultado segue de $(R^*)^{-1} = (\bigcup_{n=1}^{\infty} R^n)^{-1} = \bigcup_{n=1}^{\infty} (R^n)^{-1} = \bigcup_{n=1}^{\infty} R^n = R^*$.

25. a) $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$

c) $\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ **d)** $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

27. As respostas são as mesmas do Exercício 25. **29. a)** {(1, 1), (1, 2), (1, 4), (2, 2), (3, 3), (4, 1), (4, 2), (4, 4)} **b)** {(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (2, 4), (3, 3), (4, 1), (4, 2), (4, 4)} **c)** {(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (2, 4), (3, 3), (4, 1), (4, 2), (4, 4)} **31.** Algoritmo 1: $O(n^{3.8})$; Algoritmo 2: $O(n^3)$ **33.** Inicialize com $\mathbf{A} := \mathbf{M}_R \vee \mathbf{I}_n$ e percorra o laço apenas para $i := 2$ a $n - 1$. **35. a)** Como R é reflexiva, toda relação que a contenha também deve ser reflexiva. **b)** Ambas {(0, 0), (0, 1), (0, 2), (1, 1), (2, 2)} e {(0, 0), (0, 1), (1, 0), (1, 1), (2, 2)} contêm R e têm um número ímpar de elementos, mas nenhuma das duas é um subconjunto da outra.

Seção 8.5

1. a) Relação de equivalência. **b)** Não é reflexiva, não é transitiva. **c)** Relação de equivalência. **d)** Não é transitiva. **e)** Não é simétrica, não é transitiva. **3. a)** Relação de equivalência. **b)** Não é transitiva. **c)** Não é reflexiva, não é simétrica, não é transitiva. **d)** Relação de equivalência. **e)** Não é reflexiva, não é transitiva. **5.** São possíveis muitas respostas. (1) Dois prédios são equivalentes se eles abriram durante o mesmo ano; uma classe de equivalência consiste no conjunto dos prédios que abriram em um dado ano (contanto que exista pelo menos um prédio que abriu naquele ano). (2) Dois prédios são equivalentes se eles têm o mesmo número de andares; as classes de equivalência são os conjuntos dos prédios de 1 andar, o conjunto dos prédios de 2 andares, e assim por diante (uma classe para cada n para o qual existir pelo menos um prédio de n andares). (3) Todo prédio no qual você tiver aula é equivalente a todo prédio no qual você tiver aula (incluindo ele próprio), e todo prédio no qual você não tiver nenhuma aula é equivalente a todo prédio no qual você não tiver nenhuma aula (incluindo ele próprio); existem duas classes de equivalência — o conjunto dos prédios nos quais você tem aula e o conjunto dos prédios onde você não tem nenhuma aula (supondo que eles sejam não vazios). **7.** A afirmação "p é equivalente a q" significa que p e q têm os mesmos valores em sua tabela-verdade. R é reflexiva, pois p tem a mesma tabela-verdade que p. R é simétrica, pois, se p e q tiverem a mesma tabela-verdade, então q e p terão a mesma tabela-verdade. Se p e q tiverem os mesmos valores em suas tabelas-verdade e se q e r tiverem os mesmos valores em sua tabela-verdade, então p e r também têm, de modo que R é transitiva. A classe de equivalência de $\mathbf{T}$ é o conjunto de todas as tautologias; a classe de equivalência de $\mathbf{F}$ é o conjunto de todas as contradições. **9. a)** $(x, x) \in R$, pois $f(x) = f(x)$. Logo, R é reflexiva. $(x, y) \in R$ se e somente se $f(x) = f(y)$, o que é válido se e somente se $f(y) = f(x)$ se e somente se $(y, x) \in R$. Logo, R é simétrica. Se $(x, y) \in R$ e $(y, z) \in R$, então $f(x) = f(y)$ e $f(y) = f(z)$. Logo, $f(x) = f(z)$. Portanto, $(x, z) \in R$. Segue que R é transitiva. **b)** Os conjuntos $f^{-1}(b)$ para b na imagem de f. **11.** Seja x uma seqüência de comprimento maior que ou igual a 3. Como x coincide com ele próprio nos primeiros três bits, $(x, x) \in R$. Logo, R é reflexiva. Suponha que $(x, y) \in R$. Então x e y coincidem nos primeiros três bits. Logo, y e x coincidem nos primeiros três bits. Portanto, $(y, x) \in R$. Se (x, y) e (y, z) estiverem em R, então x e y coincidem nos primeiros três bits, do mesmo modo que y e z. Logo, x e z coincidem nos primeiros três bits. Por isso, $(x, z) \in R$. Segue que R é transitiva. **13.** Isto segue do Exercício 9, em que f é a função que leva uma seqüência de bits de comprimento maior que ou igual a 3 no par ordenado com seu primeiro bit como a primeira componente e o terceiro bit como sua segunda componente. **15.** Para reflexividade, $((a, b), (a, b)) \in R$, pois $a + b = b + a$. Para simetria, se $((a, b), (c, d)) \in R$, então $a + d = b + c$, de modo que $c + b = d + a$, assim, $((c, d), (a, b)) \in R$. Para transitividade, se $((a, b), (c, d)) \in R$ e $((c, d), (e, f)) \in R$, então $a + d = b + c$ e $c + e = d + f$, de modo que $a + d + c + e = b + c + d + f$, e assim $a + e = b + f$, e $((a, b), (e, f)) \in R$. Uma solução mais fácil é observar que, pela álgebra, a condição dada é a mesma que a condição que $f((a, b)) = f((c, d))$, em que $f((x, y)) = x - y$; portanto, pelo Exercício 9, isto é uma relação de equivalência. **17. a)** Isto segue do Exercício 9, em que a função f do conjunto das funções diferenciáveis (de $\mathbf{R}$ em $\mathbf{R}$) para o conjunto das funções (de $\mathbf{R}$ em $\mathbf{R}$) é o operador diferenciação. **b)** O conjunto de todas as funções da forma $g(x) = x^2 + C$ para alguma constante C. **19.** Isto segue do Exercício 9, em que a função f do conjunto de todas as URLs para o conjunto de todas as páginas da Web é a função que

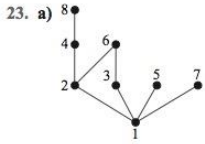
associa a cada URL a página da Web para aquela URL. **21.** Não. **23.** Não. **25.** R é reflexiva, pois uma seqüência de bits s tem o mesmo número de 1s que ela própria. R é simétrica, pois s e t ter o mesmo número de 1s implica que t e s também têm. R é transitiva, pois s e t ter o mesmo número de 1s e t e u ter o mesmo número de 1s implicam que s e u têm o mesmo número de 1s. **27. a)** Os conjuntos das pessoas da mesma idade. **b)** Os conjuntos de pessoas com os mesmos dois progenitores. **29.** O conjunto de todas as seqüências de bits com exatamente dois 1s. **31. a)** O conjunto de todas as seqüências de bits que começam com 010. **b)** O conjunto de todas as seqüências de bits que começam com 101. **c)** O conjunto de todas as seqüências de bits que começam com 111. **d)** O conjunto de todas as seqüências de bits que começam com 010. **33.** Cada uma das 15 seqüências de bits de comprimento menor que quatro está na classe de equivalência sozinha: $[\lambda]_{R_4} = \{\lambda\}$, $[0]_{R_4} = \{0\}$, $[1]_{R_4} = \{1\}$, $[00]_{R_4} = \{00\}$, $[01]_{R_4} = \{01\}$, ..., $[111]_{R_4} = \{111\}$. As 16 classes de equivalência restantes são determinadas pelas seqüências de bits de comprimento 4: $[0000]_{R_4} = \{0000, 00000, 00001, 000000, 000001, 000010, 000011, 0000000, ...\}$, $[0001]_{R_4} = \{0001, 00010, 00011, 000100, 000101, 000110, 000111, 0001000, ...\}$, ..., $[1111]_{R_4} = \{1111, 11110, 11111, 111100, 111101, 111110, 111111, 1111000, ...\}$ **35. a)** $[2]_5 = \{i | i \equiv 2 \pmod 5\} = \{..., -8, -3, 2, 7, 12, ...\}$ **b)** $[3]_5 = \{i | i \equiv 3 \pmod 5\} = \{..., -7, -2, 3, 8, 13, ...\}$ **c)** $[6]_5 = \{i | i \equiv 6 \pmod 5\} = \{..., -9, -4, 1, 6, 11, ...\}$ **d)** $[-3]_5 = \{i | i \equiv -3 \pmod 5\} = \{..., -8, -3, 2, 7, 12, ...\}$ **37.** $\{6n + k \mid n \in \mathbf{Z}\}$ para $k \in \{0, 1, 2, 3, 4, 5\}$ **39. a)** $[(1, 2)] = \{(a, b) \mid a - b = -1\} = \{(1, 2), (3, 4), (4, 5), (5, 6), ...\}$ **b)** Cada classe de equivalência pode ser interpretada como um inteiro (negativo, positivo ou zero); especificamente, $[a, b]$ pode ser interpretado como $a - b$. **41. a)** Não. **b)** Sim. **c)** Sim. **d)** Não. **43.** (a), (c), (e) **45.** (b), (d), (e) **47. a)** $\{(0, 0), (1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (3, 4), (3, 5), (4, 3), (4, 4), (4, 5), (5, 3), (5, 4), (5, 5)\}$ **b)** $\{(0, 0), (0, 1), (1, 0), (1, 1), (2, 2), (2, 3), (3, 2), (3, 3), (4, 4), (4, 5), (5, 4), (5, 5)\}$ **c)** $\{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2), (3, 3), (3, 4), (3, 5), (4, 3), (4, 4), (4, 5), (5, 3), (5, 4), (5, 5)\}$ **d)** $\{(0, 0), (1, 1), (2, 2), (3, 3), (4, 4), (5, 5)\}$ **49.** $[0]_6 \subseteq [0]_3$, $[1]_6 \subseteq [1]_3$, $[2]_6 \subseteq [2]_3$, $[3]_6 \subseteq [0]_3$, $[4]_6 \subseteq [1]_3$, $[5]_6 \subseteq [2]_3$ **51.** Seja A o conjunto na primeira partição. Escolha um elemento particular x de A. O conjunto de todas as seqüências de bits de comprimento 16 que coincidem com x nos últimos 8 bits é um dos conjuntos na segunda partição, e claramente toda seqüência em A está naquele conjunto. **53.** Afirmamos que cada classe de equivalência $[x]_{R_{31}}$ é um subconjunto da classe de equivalência $[x]_{R_8}$. Para mostrar isso, escolha um elemento arbitrário $y \in [x]_{R_{31}}$. Então, y é equivalente a x com respeito a R_{31}, de modo que ou $y = x$ ou y e x têm cada um pelo menos 31 caracteres de comprimento e coincidem em seus primeiros 31 caracteres. Como as seqüências que têm pelo menos 31 caracteres de comprimento e coincidem em seus primeiros 31 caracteres forçosamente terão pelo menos 8 caracteres de comprimento e coincidem em seus primeiros 8 caracteres, sabemos que ou $y = x$ ou y e x têm cada um pelo menos 8 caracteres de comprimento e coincidem em seus primeiros 8 caracteres. Isto significa que y é equivalente a x com respeito a R_8, de modo que $y \in [x]_{R_8}$. **55.** $\{(a, a), (a, b), (a, c), (b, a), (b, b), (b, c), (c, a), (c, b), (c, c), (d, d), (d, e), (e, d), (e, e)\}$ **57. a)** $\mathbf{Z}$ **b)** $\{n + \frac{1}{2} \mid n \in \mathbf{Z}\}$ **59. a)** R é reflexiva, pois qualquer coloração pode ser obtida de si mesma por uma rotação de 360 graus. Para ver que R é simétrica e transitiva, use o fato de que cada rotação é a composição de duas reflexões e reciprocamente a composição de duas reflexões é uma rotação. Logo, (C_1, C_2) pertence a R se e somente se C_2 puder ser obtida a partir de C_1 por uma composição de reflexões. Assim, se (C_1, C_2) pertencer a R, o mesmo ocorre com (C_2, C_1), pois a inversa da composição de reflexões também é uma composta de reflexões (na ordem inversa). Logo, R é simétrica. Para ver que R é transitiva, suponha que (C_1, C_2) e (C_2, C_3) pertençam a R. Considerar a composição das reflexões em cada caso fornece uma composição de reflexões, o que mostra que (C_1, C_3) pertence a R. **b)** Expressamos colorações com seqüências de comprimento quatro, com r e b indicando vermelho e azul, respectivamente. Listamos as letras indicando as cores do quadrado superior esquerdo, do quadrado superior direito, do quadrado inferior esquerdo e do quadrado inferior direito, nesta ordem. As classes de equivalência são: $\{rrrr\}$, $\{bbbb\}$, $\{rrrb, rrbr, rbrr, brrr\}$, $\{bbbr, bbrb, brbb, rbbb\}$, $\{rrbr, brrb\}$, $\{rrbb, brbr, bbrr, rbrb\}$. **61.** 5 **63.** Sim. **65.** R **67.** Primeiro forme o fecho reflexivo de R, então forme o fecho simétrico do fecho reflexivo, e finalmente forme o fecho transitivo do fecho simétrico do fecho reflexivo. **69.** $p(0) = 1$, $p(1) = 1$, $p(2) = 2$, $p(3) = 5$, $p(4) = 15$, $p(5) = 52$, $p(6) = 203$, $p(7) = 877$, $p(8) = 4140$, $p(9) = 21147$, $p(10) = 115975$

Seção 8.6

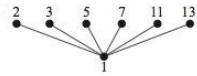
1. a) É uma ordenação parcial. **b)** Não é anti-simétrica, não é transitiva. **c)** É uma ordenação parcial. **d)** É uma ordenação parcial. **e)** Não é anti-simétrica, não é transitiva. **3. a)** Não. **b)** Não. **c)** Sim. **5. a)** Sim. **b)** Não. **c)** Sim. **d)** Não. **7. a)** Não. **b)** Sim. **c)**Não. **9.** Não. **11.** Sim. **13. a)** $\{(0, 0), (1, 0), (1, 1), (2, 0), (2, 1), (2, 2)\}$ **b)** $(\mathbf{Z}, \leq)$ **c)** $(P(\mathbf{Z}), \subseteq)$ **d)** $(\mathbf{Z}^+$, "é um múltiplo de") **15. a)** $\{0\}$ e $\{1\}$, por exemplo. **b)** 4 e 6, por exemplo. **17. a)** $(1, 1, 2) < (1, 2, 1)$ **b)** $(0, 1, 2, 3) < (0, 1, 3, 2)$ **c)** $(0, 1, 1, 1, 0) < (1, 0, 1, 0, 1)$ **19.** $0 < 0001 < 001 < 01 < 010 < 0101 < 011 < 11$

21.

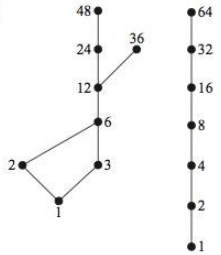
23. a)



b)



c)



25. $(a, b), (a, c), (a, d), (b, c), (b, d), (a, a), (b, b), (c, c), (d, d)$ **27.** $(a, a), (a, g), (a, d), (a, e), (a, f), (b, b), (b, g), (b, d), (b, e), (b, f), (c, c), (c, g), (c, d), (c, e), (c, f), (g, d), (g, e), (g, f), (g, g), (d, d), (e, e), (f, f)$ **29.** $(\emptyset, \{a\}), (\emptyset, \{b\}), (\emptyset, \{c\}), (\{a\}, \{a, b\}), (\{a\}, \{a, c\}), (\{b\}, \{a, b\}), (\{b\}, \{b, c\}), (\{c\}, \{a, c\}), (\{c\}, \{b, c\}), (\{a, b\}, \{a, b, c\}), (\{a, c\}, \{a, b, c\}) (\{b, c\}, \{a, b, c\})$ **31.** Seja $(S, \preccurlyeq)$ um poset finito. Mostraremos que este poset é o fecho transitivo reflexivo de sua relação de recobrimento. Suponha que (a, b) esteja no fecho transitivo reflexivo da relação de recobrimento. Então $a = b$ ou $a \prec b$, de modo que $a \preccurlyeq b$, ou então existe uma seqüência $a_1, a_2, \ldots, a_n$ tal que $a \prec a_1 \prec a_2 \prec \cdots \prec a_n \prec b$, em cujo caso novamente $a \preccurlyeq b$ pela transitividade de $\preccurlyeq$. Reciprocamente, suponha que $a \prec b$. Se $a = b$ então (a, b) está no fecho transitivo reflexivo da relação de recobrimento. Se $a \prec b$ e não existir z tal que $a \prec z \prec b$, então (a, b) é a relação de recobrimento e, portanto, em seu fecho transitivo reflexivo. Caso contrário, seja $a \prec a_1 \prec a_2 \prec \cdots \prec a_n \prec b$ a seqüência mais comprida possível desta forma (que existe, já que o poset é finito). Então, nenhum elemento intermediário pode ser inserido, de modo que cada par $(a, a_1), (a_1, a_2), \ldots, (a_n, b)$ está na relação de recobrimento, de modo que novamente (a, b) está em seu fecho transitivo reflexivo. **33. a)** 24, 45 **b)** 3,5 **c)** Não. **d)** Não. **e)** 15,45 **f)** 15 **g)** 15, 5, 3 **h)** 15 **35. a)** {1, 2}, {1, 3, 4}, {2, 3, 4} **b)** {1}, {2}, {4} **c)** Não. **d)** Não. **e)** {2, 4}, {2, 3, 4} **f)** {2, 4} **g)** {3, 4}, {4} **h)** {3, 4} **37.** Como $(a, b) \preccurlyeq (a, b)$, $\preccurlyeq$ é reflexiva. Se $(a_1, a_2) \preccurlyeq (b_1, b_2)$ e $(a_1, a_2) \neq (b_1, b_2)$, ou $a_1 \prec b_1$ ou $a_1 = b_1$ e $a_2 \prec b_2$. Em cada caso, (b_1, b_2) não é menor que ou igual a (a_1, a_2). Logo, $\preccurlyeq$ é anti-simétrica. Suponha que $(a_1, a_2) \prec (b_1, b_2) \prec (c_1, c_2)$. Então, se $a_1 \prec b_1$ ou $b_1 \prec c_1$, temos $a_1 \prec c_1$, de modo que $(a_1, a_2) \prec (c_1, c_2)$, mas, se $a_1 = b_1 = c_1$, então $a_2 \prec b_2 \prec c_2$, o que implica que $(a_1, a_2) \prec (c_1, c_2)$. Logo, $\preccurlyeq$ é transitiva. **39.** Como $(s, t) \preceq (s, t)$, $\preccurlyeq$ é reflexiva. Se $(s, t) \preccurlyeq (u, v)$ e $(u, v) \preccurlyeq (s, t)$, então $s \preccurlyeq u \preccurlyeq s$ e $t \preccurlyeq v \preccurlyeq t$; logo, $s = u$ e $t = v$. Conseqüentemente, $\preccurlyeq$ é anti-simétrica. Suponha que $(s, t) \preccurlyeq (u, v) \preccurlyeq (w, x)$. Então, $s \preccurlyeq u$, $t \preccurlyeq v$, $u \preccurlyeq w$ e $v \preccurlyeq x$. Segue que $s \preccurlyeq w$ e $t \preccurlyeq x$. Logo, $(s, t) \preccurlyeq (w, x)$. Logo, $\preccurlyeq$ é transitiva. **41. a)** Suponha que x seja maximal e que y seja o maior elemento. Então $x \preccurlyeq y$. Como x não é menor que y, segue que $x = y$. Pelo Exercício 40(a), y é único. Logo, x é único. **b)** Suponha que x seja minimal e que y seja o menor elemento. Então, $x \succcurlyeq y$. Como x não é maior que y, segue que $x = y$. Pelo Exercício 40(b), y é único. Logo, x é único. **43. a)** Sim. **b)** Não. **c)** Sim. **45.** Use indução matemática. Seja $P(n)$ a afirmação "Todo subconjunto com n elementos de um reticulado tem um menor limitante superior e um maior limitante inferior". *Passo base:* $P(1)$ é verdadeira, pois o menor limitante superior e o maior limitante inferior de $\{x\}$ são ambos x. *Passo de indução:* Suponha que $P(k)$ seja verdadeira. Seja S um conjunto com $k + 1$ elementos. Seja $x \in S$ e $S' = S - \{x\}$. Como S' tem k elementos, pela hipótese de indução, ele tem um menor limitante superior y e um maior limitante inferior a. Agora, como estamos em um reticulado, existem elementos $z = \sup(x, y)$ e $b = \inf(x, a)$. Teremos terminado se pudermos mostrar que z é o menor limitante superior de S e b é o maior limitante inferior de S. Para mostrar que z é o menor limitante superior de S, observe primeiro que, se $w \in S$, então $w = x$ ou $w \in S'$. Se $w = x$, então $w \preccurlyeq z$, pois z é o menor limitante superior de x e y. Se $w \in S'$, então $w \preccurlyeq z$, pois $w \preccurlyeq y$, o que é verdadeiro porque y é o menor limitante superior de S', e $y \preccurlyeq z$, o que é verdadeiro porque $z = \inf(x, y)$. Para ver que z é o menor limitante superior de S, suponha que u seja um limitante superior de S. Observe que tal elemento u deve ser um limitante superior de x e y, mas como $z = \inf(x, y)$, segue que $z \preccurlyeq u$. Omitimos o argumento análogo de que b é o maior limitante inferior de S. **47. a)** Não. **b)** Sim. **c)** (*Proprietário,* {*Cheetah, Puma*}), (*Restrito,* {*Cheetah, Puma*}), (*Registrado,* {*Cheetah, Puma*}), (*Proprietário,* {*Cheetah, Puma, Impala*}), (*Restrito,* {*Cheetah, Puma, Impala*}), (*Registrado,* {*Cheetah, Puma, Impala*}) **d)** (*Não proprietário,* {*Impala, Puma*}), (*Proprietário,* {*Impala, Puma*}), (*Restrito,* {*Impala, Puma*}), (*Não proprietário,* {*Impala*}), (*Proprietário,* {*Impala*}), (*Restrito,* {*Impala*}), (*Não proprietário,* {*Puma*}), (*Proprietário,* {*Puma*}), (*Restrito,* {*Puma*}), (*Não proprietário,* $\emptyset$), (*Proprietário,* $\emptyset$), (*Restrito,* $\emptyset$) **49.** Seja Π o conjunto de todas as partições de um conjunto S com $P_1 \preccurlyeq P_2$ se P_1 é um refinamento de P_2, isto é, se todo conjunto em P_1 for um subconjunto de um conjunto em P_2. Primeiro, mostramos que $(\Pi, \preccurlyeq)$ é um poset. Como $P \preccurlyeq P$ para toda partição P, $\preccurlyeq$ é reflexiva. Suponha agora que $P_1 \preccurlyeq P_2$ e $P_2 \preccurlyeq P_1$. Seja $T \in P_1$. Como $P_1 \preccurlyeq P_2$, existe um conjunto $T' \in P_2$ tal que $T \subseteq T'$. Como $P_2 \preccurlyeq P_1$, existe um conjunto $T'' \in P_1$ tal que $T' \subseteq T''$. Segue que $T \subseteq T''$. Mas como P_1 é uma partição, $T = T''$, o que implica que $T = T'$ já que $T \subseteq T' \subseteq T''$. Portanto, $T \in P_2$. Trocando os papéis de P_1 e P_2, segue que todo conjunto em P_2 também está em P_1. Logo, $P_1 = P_2$ e $\preccurlyeq$ é anti-simétrica. A seguir, suponha que $P_1 \preccurlyeq P_2$ e $P_2 \preccurlyeq P_3$. Seja $T \in P_1$. Então, existe um conjunto $T' \in P_2$ tal que $T \subseteq T'$. Como $P_2 \preccurlyeq P_3$, existe um conjunto $T'' \in P_3$ tal que $T' \subseteq T''$. Isto significa que $T \subseteq T''$. Logo, $P_1 \preccurlyeq P_3$. Segue que $\preccurlyeq$ é transitiva. O maior limitante inferior das partições P_1 e P_2 é a partição P, cujos subconjuntos são os conjuntos não vazios da forma $T_1 \cap T_2$, em que $T_1 \in P_1$ e $T_2 \in P_2$. Omitimos aqui a justificativa desta afirmação. O menor limitante superior das partições P_1 e P_2 é a partição que corresponde à relação de equivalência na qual $x \in S$ está relacionado a $y \in S$ se existir uma seqüência $x = x_0, x_1, x_2, \ldots, x_n = y$ para algum inteiro não negativo n tal que para cada i de 1 a n, x_{i-1} e x_i estão no mesmo elemento de P_1 ou de P_2. Omitiremos os detalhes de que isto é uma relação de equivalência e os detalhes da demonstração de que isto é o menor limitante superior das duas partições. **51.** Pelo Exercício 45, existe um menor limitante superior e um maior limitante inferior para o todo reticulado finito. Por definição, estes ele-

mentos são o maior e o menor elementos, respectivamente. **53.** O menor elemento de um subconjunto de $\mathbf{Z}^+ \times \mathbf{Z}^+$ é o par que tem a menor primeira coordenada possível e, se existir mais de um destes pares, o par entre eles que tiver a menor segunda coordenada. **55.** Se x for um inteiro em uma seqüência decrescente de elementos neste poset, então no máximo $|x|$ elementos podem seguir x na seqüência, ou seja, inteiros cujos valores absolutos são $|x|-1$, $|x|-2$, ..., 1, 0. Portanto, não pode haver nenhuma seqüência decrescente infinita. Este não é um conjunto totalmente ordenado, pois 5 e -5, por exemplo, não são comparáveis. **57.** Para encontrar qual entre dois números racionais é maior, escreva-os com um denominador comum positivo e compare os numeradores. Para mostrar que este conjunto é denso, suponha que $x < y$ sejam dois números racionais. Então, sua média, isto é, $(x+y)/2$, é um número racional entre eles. **59.** Seja $(S, \preccurlyeq)$ um conjunto parcialmente ordenado. É suficiente mostrar que todo subconjunto não vazio de S contém um menor elemento se e somente se não existirem seqüências decrescentes infinitas de elementos $a_1, a_2, a_3, \ldots$ em S (isto é, nas quais $a_{i+1} \prec a_i$ para todo i). Uma seqüência decrescente infinita de elementos claramente não tem um menor elemento. Reciprocamente, seja A um subconjunto não vazio qualquer de S que não tenha um menor elemento. Como A é não vazio, escolha $a_1 \in A$. Como a_1 não é o menor elemento de A, escolha $a_2 \in A$, com $a_2 \prec a_1$. Como a_2 não é o menor elemento de A, escolha $a_3 \in A$, com $a_3 \prec a_2$. Continue desta maneira, produzindo uma seqüência decrescente infinita em S. **61.** $a \prec_t b \prec_t c \prec_t d \prec_t e \prec_t f \prec_t g \prec_t h \prec_t i \prec_t j \prec_t k \prec_t l \prec_t m$ **63.** $C \prec A \prec B \prec D \prec E \prec F \prec G$ **65.** Determinar as necessidades do usuário $\prec$ Escrever os requisitos funcionais $\prec$ Montar locais de teste $\prec$ Desenvolver os requisitos do sistema $\prec$ Escrever documentação $\prec$ Desenvolver o módulo $A \prec$ Desenvolver o módulo $B \prec$ Desenvolver o módulo $C \prec$ Integrar os módulos $\prec$ Teste $\alpha \prec$ Teste $\beta \prec$ Acabamento

Exercícios Complementares

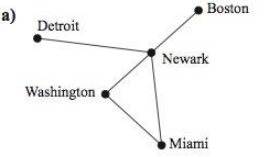
1. a) Irreflexiva (não incluímos a seqüência vazia), simétrica. **b)** Irreflexiva, simétrica **c)** Irreflexiva, anti-simétrica, transitiva. **3.** $((a,b),(a,b)) \in R$, já que $a+b=a+b$. Logo, R é reflexiva. Se $((a,b),(c,d)) \in R$, então $a+d=b+c$, de modo que $c+b=d+a$. Segue que $((c,d),(a,b)) \in R$. Logo, R é simétrica. Suponha que $((a,b),(c,d))$ e $((c,d),(e,f))$ pertençam a R. Então $a+d=b+c$ e $c+f=d+e$. Somando estas duas equações e subtraindo $c+d$ de ambos os lados, obtemos $a+f=b+e$. Logo, $((a,b),(e,f))$ pertencem a R. Logo, R é transitiva. **5.** Suponha que $(a,b) \in R$. Como $(b,b) \in R$, segue que $(a,b) \in R^2$. **7.** Sim, sim. **9.** Sim, sim. **11.** Dois registros com chaves idênticas na projeção teriam chaves idênticas no original. **13.** $(\Delta \cup R)^{-1} = \Delta^{-1} \cup R^{-1} = \Delta \cup R^{-1}$ **15. a)** $R = \{(a,b),(a,c)\}$. O fecho transitivo do fecho simétrico de R é $\{(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a),(c,b),(c,c)\}$ e é diferente do fecho simétrico do fecho transitivo de R, que é $\{(a,b),(a,c),(b,a),(c,a)\}$. **b)** Suponha que (a,b) esteja no fecho simétrico do fecho transitivo de R. Devemos mostrar que (a,b) está no fecho transitivo do fecho simétrico de R. Sabemos que pelo menos um entre (a,b) e (b,a) está no fecho transitivo de R. Logo, existe ou um caminho de a a b em R ou um caminho de b a a em R (ou ambos). No primeiro caso, existe um caminho de a a b no fecho simétrico de R. No último caso, podemos formar um caminho de a a b no fecho simétrico de R invertendo os sentidos de todas as arestas em um caminho de b a a, indo para trás. Logo, (a,b) está no fecho transitivo do fecho simétrico de R. **17.** O fecho de S com relação à propriedade **P** é a relação com a propriedade **P** que contém R, pois $R \subseteq S$. Logo, o fecho de S com relação à propriedade **P** contém o fecho de R com relação à propriedade **P**. **19.** Use a idéia básica do algoritmo de Warshall, exceto pelo fato de considerar $w_{ij}^{[k]}$ igual ao comprimento do caminho mais longo de v_i a v_j usando vértices interiores com subscritos que não excedam k, e igual a -1 se não existir tal caminho. Para encontrar $w_{ij}^{[k]}$ a partir dos elementos de $\mathbf{W}_{k-1}$, determine para cada par (i,j) se existem caminhos de v_i a v_k e de v_k a v_j sem usar nenhum vértice rotulado maior que k. Se $w_{ik}^{[k-1]}$ ou $w_{kj}^{[k-1]}$ é -1, então tal par de caminhos não existe, de modo que faça $w_{ij}^{[k]} = w_{ij}^{[k-1]}$. Se tal par de caminhos existir, então há duas possibilidades. Se $w_{kk}^{[k-1]} > 0$, existem caminhos arbitrariamente longos de v_i a v_j, de modo que faça $w_{ij}^{[k]} = \infty$. Se $w_{kk}^{[k-1]} = 0$, Considere $w_{ij}^{[k-1]} = \max(w_{ij}^{[k-1]}, w_{ik}^{[k-1]} + w_{kj}^{[k-1]})$. (Considere inicialmente $\mathbf{W}_0 = \mathbf{M}_R$.) **21.** 25. **23.** Como $A_i \cap B_j$ é um subconjunto de A_i e de B_j, a coleção dos subconjuntos é um refinamento de cada uma das partições dadas. Devemos mostrar que ela é uma partição. Por construção, cada um destes conjuntos é não vazio. Para ver que a união deles é S, suponha que $s \in S$. Como P_1 e P_2 são partições de S, existem conjuntos A_i e B_j tais que $s \in A_i$ e $s \in B_j$. Portanto, $s \in A_i \cap B_j$. Logo, a união destes conjuntos é S. Para ver que eles são dois a dois disjuntos, observe que a menos que $i = i'$ e $j = j'$, $(A_i \cap B_j) \cap (A_{i'} \cap B_{j'}) = (A_i \cap A_{i'}) \cap (B_j \cap B_{j'}) = \emptyset$. **25.** A relação subconjunto é uma ordem parcial em qualquer coleção de conjuntos, pois ela é reflexiva, anti-simétrica e transitiva. Aqui, a coleção de conjuntos é $\mathbf{R}(S)$. **27.** Achar receita $\prec$ Comprar frutos do mar $\prec$ Comprar artigos de mercearia $\prec$ Lavar os mariscos $\prec$ Cortar o gengibre e o alho $\prec$ Limpar os peixes $\prec$ Cozinhar o arroz $\prec$ Cortar os peixes $\prec$ Lavar os vegetais $\prec$ Cortar castanhas $\prec$ Fazer as guarnições $\prec$ Cozinhar $\prec$ Arrumar nos pratos $\prec$ Servir. **29. a)** A única anticadeia com mais de um elemento é $\{c,d\}$. **b)** As únicas anticadeias com mais de um elemento são $\{b,c\}$, $\{c,e\}$ e $\{d,e\}$. **c)** As únicas anticadeias com mais de um elemento são $\{a,b\}$, $\{a,c\}$, $\{b,c\}$, $\{a,b,c\}$, $\{d,e\}$, $\{d,f\}$, $\{e,f\}$ e $\{d,e,f\}$. **31.** Seja $(S, \preccurlyeq)$ um poset finito, e seja A uma cadeia maximal. Como $(A, \preccurlyeq)$ também é um poset, ela deve ter um elemento mínimal m. Suponha que m não seja mínimal em S. Então, existiria um elemento a em S com $a \prec m$. Entretanto, isto tornaria o conjunto $A \cup \{a\}$ uma cadeia maior que A. Para mostrar isto, devemos mostrar que a é comparável a todo elemento de A. Como m é comparável a todo elemento de A e m é mínimal, segue que $m \prec x$ quando x está em A e $x \neq m$. Como $a \prec m$ e $m \prec x$, a propriedade transitiva mostra que $a \prec x$ para todo elemento de A. **33.** Indique por $a\ Rb$ o fato de a ser um descendente de b. Pelo Exercício 32, se não existir nenhum conjunto de $n+1$ pessoas em que nenhuma das quais é um descendente de nenhuma outra (uma anticadeia), então $k \leq n$, de modo que o conjunto pode ser dividido em $k \leq n$ cadeias. Pelo princípio da casa dos pombos, pelo menos uma destas

cadeias contém no mínimo $m + 1$ pessoas. **35.** Demonstramos por contradição que, se S não tiver nenhuma seqüência decrescente infinita e $\forall x\,(\{\forall y[y \prec x \rightarrow P(y)]\} \rightarrow P(x))$, então $P(x)$ é verdadeira para todo $x \in S$. Se não for válido que $P(x)$ é verdadeira para todo $x \in S$, seja x_1 um elemento de S tal que $P(x_1)$ não seja verdadeira. Então, pela afirmação condicional já dada, deve ocorrer que $\forall y[y \prec x_1 \rightarrow P(y)]$ não é verdadeira. Isto significa que existe algum x_2 com $x_2 \prec x_1$ tal que $P(x_2)$ não é verdadeira. Novamente, invocando a afirmação condicional, obtemos um $x_3 \prec x_2$ tal que $P(x_3)$ não é verdadeira, e assim por diante indefinidamente. Isto contradiz o fato de nosso poset ser bem fundado. Portanto, $P(x)$ é verdadeira para todo $x \in S$. **37.** Suponha que R seja uma quase-ordem. Como R é reflexiva, se $a \in A$, então $(a, a) \in R$. Isto implica que $(a, a) \in R^{-1}$. Logo, $a \in R \cap R^{-1}$. Segue que $R \cap R^{-1}$ é reflexiva. $R \cap R^{-1}$ é simétrica para qualquer relação R, pois, para qualquer relação R, se $(a, b) \in R$, então $(b, a) \in R^{-1}$ e vice-versa. Para mostrar que $R \cap R^{-1}$ é transitiva, suponha que $(a, b) \in R \cap R^{-1}$ e $(b, c) \in R \cap R^{-1}$. Como $(a, b) \in R$ e $(b, c) \in R$, $(a, c) \in R$, pois R é transitiva. Analogamente, como $(a, b) \in R^{-1}$ e $(b, c) \in R^{-1}$, $(b, a) \in R$ e $(c, b) \in R$, de modo que $(c, a) \in R$ e $(a, c) \in R^{-1}$. Logo, $(a, c) \in R \cap R^{-1}$. Segue que $R \cap R^{-1}$ é uma relação de equivalência. **39. a)** Como $\inf(x, y) = \inf(y, x)$ e $\sup(x, y) = \sup(y, x)$, segue que $x \wedge y = y \wedge x$ e $x \vee y = y \vee x$. **b)** Usando a definição, $(x \wedge y) \wedge z$ é um limitante inferior de x, y e z que é maior que qualquer outro limitante inferior. Como x, y e z têm papéis intercambiáveis, $x \wedge (y \wedge z)$ é o mesmo elemento. Analogamente, $(x \vee y) \vee z$ é um limitante superior de x, y e z que é menor que qualquer outro limitante superior. Como x, y e z têm papéis intercambiáveis, $x \vee (y \vee z)$ é o mesmo elemento. **c)** Para mostrar que $x \wedge (x \vee y) = x$, é suficiente mostrar que x é o maior limitante inferior de x, e $x \vee y$. Observe que x é um limitante inferior de x, e, como $x \vee y$ é por definição maior que x, x é um limitante inferior para ele também. Portanto, x é um limitante inferior. Mas qualquer limitante inferior de x deve ser menor que x, de modo que x é o maior limitante inferior. A segunda afirmação é a dual da primeira; omitimos sua demonstração. **d)** x é um limitante inferior e um limitante superior para si mesmo e o maior e o menor de tais limitantes. **41. a)** Como 1 é o único elemento maior ou igual a 1, ele é o único limitante superior para 1 e, portanto, o único valor possível do menor limitante superior de x e 1. **b)** Como $x \preccurlyeq 1$, x é um limitante inferior para ambos, x e 1, e nenhum outro limitante inferior pode ser maior que x, de modo que $x \wedge 1 = x$. **c)** Como $0 \preccurlyeq x$, x é um limitante superior para ambos, x e 0, e nenhum outro limitante pode ser menor que x, de modo que $x \vee 0 = x$. **d)** Como 0 é o único elemento menor que ou igual a 0, ele é o único limitante inferior para 0 e, portanto, o único valor possível do maior limitante inferior de x e 0. **43.** $L = (S, \subseteq)$ em que $S = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{1, 2, 3\}\}$ **45.** Sim. **47.** O complemento de um subconjunto $X \subseteq S$ é seu complemento $S - X$. Para demonstrar isto, observe que $X \vee (S - X) = 1$ e $X \wedge (S - X) = 0$, já que $X \cup (S - X) = S$ e $X \cap (S - X) = \emptyset$. **49.** Pense na grade retangular como representando elementos em uma matriz. Portanto, numeramos de cima para baixo e, dentro disso, da esquerda para a direita. A ordem parcial é que $(a, b) \preceq (c, d)$ se e somente se $a \leq c$ e $b \leq d$. Observe que $(1, 1)$ é o menor elemento com respeito a esta relação. As regras para o Chomp, como explicado no Capítulo 1, coincidem com as regras enunciadas aqui no preâmbulo. Mas agora podemos identificar o ponto (a, b) com o número natural $p^{a-1}q^{b-1}$ para todo a e b com $1 \leq a \leq m$ e $1 \leq b \leq n$. Isto identifica os pontos na grade retangular com o conjunto S neste exercício, e a ordem parcial $\preceq$ que acabamos de descrever é a mesma que a relação divide, pois $p^{a-1}q^{b-1} \mid p^{c-1}q^{d-1}$ se e somente se o expoente em p na esquerda não exceder o expoente de p na direita, e analogamente para q.

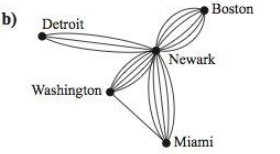
CAPÍTULO 9

Seção 9.1

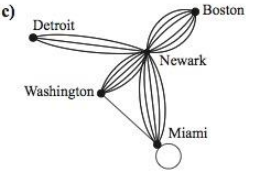
1. a)



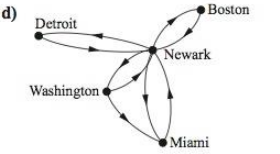
b)



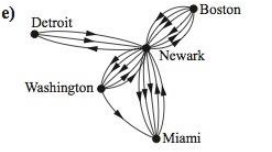
c)



d)

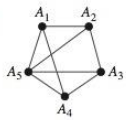


e)

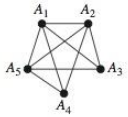


3. Grafo simples. **5.** Pseudografo. **7.** Grafo orientado. **9.** Multigrafo orientado. **11.** Se $u\ Rv$, então existe uma aresta associada com $\{u, v\}$. Mas $\{u, v\} = \{v, u\}$, de modo que esta aresta está associada a $\{v, u\}$ e, portanto, $v\ Ru$. Assim, por definição, R é uma relação simétrica. Um grafo simples não permite laços; desse modo, $u\ Ru$ nunca é verdadeira, e assim, pela definição, R é irreflexiva.

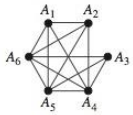
13. a)



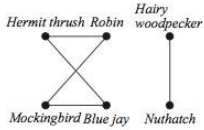
b)



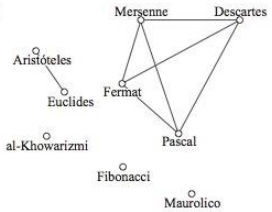
c)

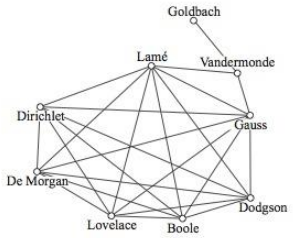


15.

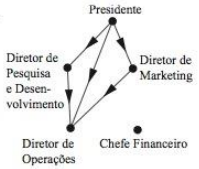


17.

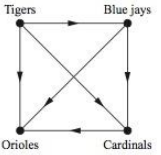




19.

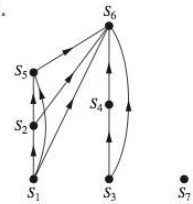


21.



23. Encontramos os números de telefone no grafo de chamadas de fevereiro que não estão presentes no grafo de chamadas de janeiro e vice-versa. Para cada número que encontramos, fazemos uma lista dos números que elas chamaram ou que as chamaram usando as arestas no grafo de chamadas. Examinamos esta lista para encontrar novos números de telefone em fevereiro, que tinham padrões de chamadas parecidos com números de telefone extintos em janeiro. **25.** Usamos o modelo de grafo que tem endereços de e-mail como vértices e para cada mensagem enviada, uma aresta do endereço de e-mail de quem enviou para o endereço de e-mail de quem recebeu a mensagem. Para cada endereço de e-mail, podemos fazer uma lista de outros endereços para os quais eles mandaram mensagens e uma lista de outros endereços dos quais eles receberam mensagens. Se dois endereços de e-mail têm quase o mesmo padrão, concluímos que estes endereços poderiam pertencer à mesma pessoa que recentemente mudou seu endereço de e-mail. **27.** Seja V o conjunto das pessoas na festa. Seja E o conjunto dos pares ordenados (u, v) em $V \times V$, tal que u sabe o nome de v. As arestas são orientadas, mas arestas múltiplas não são permitidas. Literalmente, existe um laço em cada vértice, mas por simplicidade, o modelo poderia omitir os laços. **29.** Considere o conjunto de vértices como o conjunto de pessoas, e dois vértices são ligados por uma aresta se as duas pessoas já foram casadas. Ignorando complicações, este grafo tem a propriedade de que existem dois tipos de vértices (homens e mulheres), e cada aresta liga vértices de tipos opostos.

31.



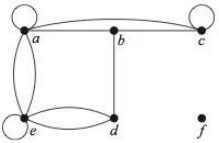
33. Represente as pessoas no grupo por vértices. Ponha uma aresta orientada no grafo para cada par de vértices. Rotule a aresta do vértice que representa A até o vértice que representa B com um $+$ (mais) se A gostar de B, um $-$ (menos) se A não gostar de B, e com um 0 se A for neutro em relação a B.

Seção 9.2

1. $v = 6$; $e = 6$; $\text{gr}(a) = 2$, $\text{gr}(b) = 4$, $\text{gr}(c) = 1$, $\text{gr}(d) = 0$, $\text{gr}(e) = 2$, $\text{gr}(f) = 3$; c é pendente; d é isolado. **3.** $v = 9$; $e = 12$; $\text{gr}(a) = 3$, $\text{gr}(b) = 2$, $\text{gr}(c) = 4$, $\text{gr}(d) = 0$, $\text{gr}(e) = 6$, $\text{gr}(f) = 0$; $\text{gr}(g) = 4$; $\text{gr}(h) = 2$; $\text{gr}(i) = 3$; d e f são isolados. **5.** Não, pois a soma dos graus dos vértices não

pode ser ímpar. **7.** $v = 4$; $e = 7$; $\text{gr}^-(a) = 3$, $\text{gr}^-(b) = 1$, $\text{gr}^-(c) = 2$, $\text{gr}^-(d) = 1$, $\text{gr}^+(a) = 1$, $\text{gr}^+(b) = 2$, $\text{gr}^+(c) = 1$, $\text{gr}^+(d) = 3$ **9.** 5 vértices, 13 arestas; $\text{gr}^-(a) = 6$, $\text{gr}^+(a) = 1$, $\text{gr}^-(b) = 1$, $\text{gr}^+(b) = 5$, $\text{gr}^-(c) = 2$, $\text{gr}^+(c) = 5$, $\text{gr}^-(d) = 4$, $\text{gr}^+(d) = 2$, $\text{gr}^-(e) = 0$, $\text{gr}^+(e) = 0$

11.



13. O número de colaboradores que v tem; alguém que nunca teve colaboradores; alguém que tenha apenas um colaborador. **15.** Nenhum grafo orientado, $\text{gr}^-(v)$ = número de chamadas que v recebeu, $\text{gr}^+(v)$ = número de chamadas que v fez; no grafo não orientado, $\text{gr}(v)$ é o número de chamadas feitas ou recebidas por v. **17.** $(\text{gr}^+(v), \text{gr}^-(v))$ é o registro de ganhos e perdas de v. **19.** Construa o modelo de grafo simples no qual V é o conjunto de pessoas no grupo e no qual existe uma aresta associada a $\{u, v\}$ se u e v se conhecem. Então, o grau do vértice v é o número de pessoas que v conhece. Pelo resultado do Exercício 18, existem dois vértices com o mesmo grau. Portanto, existem duas pessoas que conhecem o mesmo número de outras pessoas no grupo. **21.** Bipartido. **23.** Não é bipartido. **25.** Não é bipartido. **27. a)** Seja $V = \{$Zamora, Agraharam, Smith, Chou, Macintyre, planejamento, publicidade, vendas, propaganda, desenvolvimento, relações na indústria$\}$ e $E = \{\{$Zamora, planejamento$\}$, $\{$Zamora, vendas$\}$, $\{$Zamora, propaganda$\}$, $\{$Zamora, relações na indústria$\}$, $\{$Agraharam, planejamento$\}$, $\{$Agraharam, desenvolvimento$\}$, $\{$Smith, publicidade$\}$, $\{$Smith, vendas$\}$, $\{$Smith, relações na indústria$\}$, $\{$Chou, planejamento$\}$, $\{$Chou, vendas$\}$, $\{$Chou, relações na indústria$\}$, $\{$Macintyre, planejamento$\}$, $\{$Macintyre, publicidade$\}$, $\{$Macintyre, vendas$\}$, $\{$Macintyre, relações na indústria$\}\}$. **b)** Muitas respostas são possíveis, tal como Zamora–planejamento, Agraharam–desenvolvimento, Smith–publicidade, Chou–vendas e Macintyre–relações na indústria. **29. a)** n vértices, $n(n-1)/2$ arestas **b)** n vértices, n arestas **c)** $n+1$ vértices, $2n$ arestas **d)** $m+n$ vértices, mn arestas **e)** 2^n vértices, $n2^{n-1}$ arestas. **31. a)** 3, 3, 3, 3 **b)** 2, 2, 2, 2 **c)** 4, 3, 3, 3, 3 **d)** 3, 3, 2, 2, 2 **e)** 3, 3, 3, 3, 3, 3, 3 **33.** Cada um dos n vértices é adjacente a cada um dos outros $n-1$ vértices, de modo que a seqüência de graus é $n-1, n-1, ..., n-1$(n termos). **35.** 7

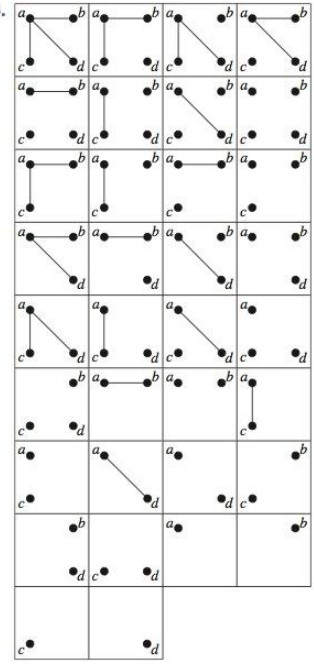
37. a) Sim

b) Não, a soma dos graus é ímpar. **c)** Não. **d)** Não, a soma dos graus é ímpar.

e) Sim.

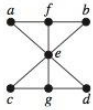
f) Não, a soma dos graus é ímpar. **39.** Primeiro, suponha que $d_1, d_2, ..., d_n$ seja gráfica. Precisamos mostrar que a seqüência cujos termos são $d_2 - 1, d_3 - 1, ..., d_{d_1+1} - 1, d_{d_1+2}, d_{d_1+3}, ..., d_n$ é gráfica uma vez que tenha sido posta em ordem não crescente. No Exercício 38 é demonstrado que se a seqüência original for gráfica, então de fato existe um grafo que tem esta seqüência de graus na qual cada vértice de grau d_1 é adjacente aos vértices de graus $d_2, d_3, ..., d_{d_1+1}$. Remova deste grafo o vértice de grau mais alto (d_1). O grafo resultante tem a seqüência de graus desejada. Reciprocamente, suponha que $d_1, d_2, ..., d_n$ seja uma seqüência não crescente, tal que a seqüência $d_2 - 1, d_3 - 1, ..., d_{d_1+1} - 1, d_{d_1+2}, d_{d_1+3}, ..., d_n$ seja gráfica uma vez que tenha sido posta em ordem não crescente. Considere um grafo com esta última seqüência de graus, em que o vértice v_i tem grau $d_i - 1$ para $2 \le i \le d_1 + 1$ e o vértice v_i tem grau d_i para $d_1 + 2 \le i \le n$. Adicione um novo vértice (chame-o de v_1), e ponha uma aresta de v_1 para cada um dos vértices $v_2, v_3, ..., v_{d_1+1}$. O grafo resultante tem uma seqüência de graus $d_1, d_2, ..., d_n$. **41.** Seja $d_1, d_2, ..., d_n$ uma seqüência não crescente de inteiros não negativos com soma par. Construa um grafo como a seguir: Considere os vértices $v_1, v_2, ..., v_n$ e ponha $\lfloor d_i/2 \rfloor$ laços no vértice v_i, para $i = 1, 2, ..., n$. Para cada i, o vértice v_i agora tem grau d_i ou $d_i - 1$. Como a soma original era par, o número de vértices para os quais $\text{gr}(v_i) = d_i - 1$ é par. Emparelhe-os arbitrariamente e coloque uma aresta ligando os vértices em cada par.

43.



47. **a)** Para todo $n \geq 1$ **b)** Para todo $n \geq 3$ **c)** Para $n = 3$ **d)** Para todo $n \geq 0$ **49.** 5

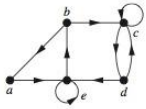
51.



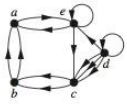
53. **a)** O grafo com n vértices e nenhuma aresta. **b)** A união disjunta de K_m e K_n **c)** O grafo com vértices $\{v_1, ..., v_n\}$ com uma aresta entre v_i e v_j a menos que $i \equiv j \pm 1 \pmod{n}$. **d)** O grafo cujos vértices são representados por seqüências de bits de comprimento n com uma aresta entre dois vértices se as seqüências de bits associadas diferirem em mais de um bit.

55. $v(v-1)/2 - e$ **57.** $n-1-d_n, n-1-d_{n-1}, ..., n-1-d_2, n-1-d_1$. **59.** A união de G e $\overline{G}$ contém uma aresta entre cada par de n vértices. Logo, esta união é K_n.

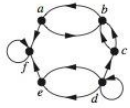
61. Exercício 7:



Exercício 8:

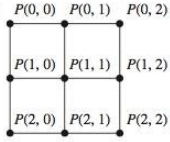


Exercício 9:



63. Um grafo orientado $G = (V, E)$ é seu próprio reverso se e somente se ele satisfaz a condição $(u, v) \in E$ se e somente se $(v, u) \in E$. Mas esta é precisamente a condição que a relação associada deve satisfazer para ser simétrica.

65.



67. Podemos conectar $P(i, j)$ e $P(k, l)$ usando $|i-k|$ hops para conectar $P(i, j)$ e $P(k, j)$ e $|j-l|$ hops para conectar $P(k, j)$ e $P(k, l)$. Logo, o número total de hops necessário para conectar $P(i, j)$ e $P(k, l)$ não excede $|i-k|+|j-l|$. isto é menor que ou igual a $m + m = 2m$, que é $O(m)$.

Seção 9.3

1.

Vértice	*Vértices Adjacentes*
a	*b, c, d*
b	*a, d*
c	*a, d*
d	*a, b, c*

3.

Vértice	*Vértices Terminais*
a	*a, b, c, d*
b	*d*
c	*a, b*
d	*b, c, d*

5.
$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

em que os vértices estão listados em ordem alfabética.

7.
$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

9. **a)**
$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

b)
$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

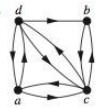
c)
$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

d)
$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

e)
$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

f)
$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

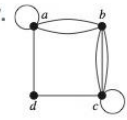
11.



13.
$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

15.
$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}$$

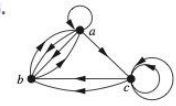
17.



19.
$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

21.
$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

23.



25. sim

27. Exercício 13: $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$

Exercício 14: $\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$

Exercício 15: $\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$

29. $\text{gr}(v)$ – número de laços em v; $\text{gr}(v)$ **31.** 2 se e não for um laço, 1 se e for um laço

33. a) $\begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 1 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 1 & 1 \end{bmatrix}$

c) $\begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\ & & & & 1 & 0 & \cdots & 0 \\ & & \mathbf{B} & & 0 & 1 & \cdots & 0 \\ & & & & \vdots & \vdots & & \vdots \\ & & & & 0 & 0 & \cdots & 1 \end{bmatrix}$

em que **B** é a resposta de (b)

d) $\begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \\ 1 & 0 & & 0 & 1 & \cdots & 0 \\ 0 & 1 & & 0 & 0 & & \\ \vdots & \vdots & & \vdots & \vdots & & \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 1 \end{bmatrix}$

35. Isomorfo **37.** Isomorfo **39.** Isomorfo **41.** Não isomorfo **43.** Isomorfo **45.** G é isomorfo a ele próprio pela função identidade, de modo que isomorfismo é reflexivo. Suponha que G seja isomorfo a H. Então, existe uma correspondência biunívoca f de G em H que preserva adjacência e não adjacência. Segue que f^{-1} é uma correspondência biunívoca de H em G que preserva adjacência e não adjacência. Logo, isomorfismo é símétrico. Se G for isomorfo a H e H for isomorfo a K, então existem correspondências biunívocas f e g de G em H e de H em K que preservam adjacência e não adjacência. Segue que $g \circ f$ é uma correspondência biunívoca de G em K que preserva adjacência e não adjacência. Logo, isomorfismo é transitivo. **47.** Todos zeros. **49.** Rotule os vértices em ordem, de modo que todos os vértices no primeiro conjunto da partição do conjunto de vértices venham primeiro. Como nenhuma aresta liga vértices no mesmo conjunto da partição, a matriz tem a forma pedida. **51.** C_5 **53.** $n = 5$ apenas. **55.** 4 **57. a)** Sim. **b)** Não. **c)** Não. **59.** $G = (V_1, E_1)$ é isomorfo a $H = (V_2, E_2)$ se e somente se existirem funções f de V_1 em V_2 e g de E_1 em E_2, tal que cada uma é uma correspondência biunívoca e para toda aresta e em E_1 as extremidades de $g(e)$ são $f(v)$ e $f(w)$ em que v e w são as extremidades de e. **61.** Sim. **63.** Sim. **65.** Se f for um isomorfismo de um grafo orientado G para um grafo orientado H, então f também é um isomorfismo de G^{conv} em H^{conv}. Para perceber isso, observe que (u, v) é uma aresta de G^{conv} se e somente se (v, u) é uma aresta de G se e somente se $(f(v), f(u))$ é uma aresta de H se e somente se $(f(u), f(v))$ é uma aresta de H^{conv}. **67.** São possíveis muitas respostas; por exemplo, C_6 e $C_3 \cup C_3$. **69.** O produto é $[a_{ij}]$, em que a_{ij} é o número de arestas de v_i para v_j quando $i \neq j$ e a_{ii} é o número de arestas incidentes a v_i. **71.** Os grafos do Exercício 41 fornecem um par do diabo.

Seção 9.4

1. a) Caminho de comprimento 4; não é um ciclo; não é simples. **b)** Não é um caminho. **c)** Não é um caminho. **d)** Ciclo simples de comprimento 5. **3.** Não. **5.** Não. **7.** Conjuntos maximais de pessoas com a propriedade de que para quaisquer dois deles, podemos encontrar uma seqüência de conhecidos que nos leva de um para outro. **9.** Se uma pessoa tem o número de Erdős n, então existe um caminho de comprimento n desta pessoa até Erdős no grafo de colaboração, de modo que, por definição, isto significa que a pessoa está na mesma componente Erdős. Se a pessoa estiver na mesma componente que Erdős, então existe um caminho daquela pessoa até Erdős, e o comprimento do caminho mais curto entre estes é o número de Erdős da pessoa. **11. a)** Fracamente conexo. **b)** Fracamente conexo. **c)** Nem fortemente nem fracamente conexo. **13.** Os conjuntos maximais de números de telefone para os quais é possível encontrar caminhos orientados entre quaisquer dois números diferentes no conjunto. **15. a)** $\{a, b, f\}$, $\{c, d, e\}$ **b)** $\{a, b, c, d, e, h\}$, $\{f\}$, $\{g\}$ **c)** $\{a, b, d, e, f, g, h, i\}$, $\{c\}$ **17. a)** 2 **b)** 7 **c)** 20 **d)** 61 **19.** Não isomorfo (G tem um triângulo; H não). **21.** Isomorfo (o caminho $u_1, u_2, u_7, u_6, u_5, u_4, u_3, u_8, u_1$ corresponde ao caminho $v_1, v_2, v_3, v_4, v_5, v_8, v_7, v_6, v_1$) **23. a)** 3 **b)** 0 **c)** 27 **d)** 0 **25. a)** 1 **b)** 0 **c)** 2 **d)** 1 **e)** 5 **f)** 3 **27.** R é reflexiva por definição. Suponha que $(u, v) \in R$; então existe um caminho de u a v. Então, $(v, u) \in R$, pois existe um caminho de v a u, ou seja, o caminho de u a v percorrido de trás para frente. Suponha que $(u, v) \in R$ e $(v, w) \in R$; então existem caminhos de u a v e de v a w. Juntando estes dois caminhos, obtemos um caminho de u a w. Logo, $(u, w) \in R$. Segue que R é transitiva. **29.** c **31.** b, c, e, i **33.** Se um vértice é

pendente, ele claramente não é um vértice de corte. Assim, a extremidade de uma aresta de corte que seja um vértice de corte não é pendente. A remoção de uma aresta de corte produz um grafo com mais componentes conexas que o grafo original. Se uma extremidade de uma aresta de corte não é pendente, a componente conexa na qual ela está depois da remoção da aresta de corte contém mais que apenas este vértice. Conseqüentemente, a remoção daquele vértice e de todas as arestas incidentes a ele, incluindo a aresta de corte original, produz um grafo com mais componentes conexas que o grafo original. Logo, uma extremidade de uma aresta de corte que não seja pendente é um vértice de corte. **35.** Suponha que exista um grafo conexo G com no máximo um vértice que não seja um vértice de corte. Defina a distância entre os vértices u e v, indicada por $d(u, v)$, como o comprimento do caminho mais curto entre u e v em G. Sejam s e t vértices em G, tal que $d(s, t)$ seja um máximo. Ou s ou t (ou ambos) é um vértice de corte, assim, sem perda de generalidade, suponha que s seja um vértice de corte. Seja w pertencente à componente conexa que não contém t do grafo obtido pela remoção de s e de todas as arestas incidentes a s a partir de G. Como todo caminho de w a t contém s, $d(w, t) > d(s, t)$, o que é uma contradição. **37. a)** Denver–Chicago, Boston–Nova York **b)** Seattle–Portland, Portland–São Francisco, Salt Lake City–Denver, Nova York–Boston, Boston–Burlington, Boston–Bangor. **39.** Um conjunto de pessoas que coletivamente influenciam todo mundo (direta ou indiretamente); {Deborah} **41.** Uma aresta não pode ligar dois vértices em componentes conexas diferentes. Como existem no máximo $C(n_i, 2)$ arestas na componente conexa com n_i vértices, segue que existem no máximo $\sum_{i=1}^{k} C(n_i, 2)$ arestas no grafo. **43.** Suponha que G não seja conexo. Então, ele tem uma componente com k vértices para algum k, $1 \leq k \leq n-1$. O máximo de arestas que G poderia ter é $C(k, 2) + C(n-k, 2) = [k(k-1) + (n-k)(n-k-1)]/2 = k^2 - nk + (n^2 - n)/2$. Esta função quadrática de f é minimizada em $k = n/2$ e maximizada em $k = 1$ ou $k = n-1$. Logo, se G não for conexo, o número de arestas não excede o valor desta função em 1 e em $n-1$, ou seja, $(n-1)(n-2)/2$. **45. a)** 1 **b)** 2 **c)** 6 **d)** 21 **47.** 2 **49.** Sejam P_1 e P_2 os caminhos $u = x_0, x_1, \ldots, x_n = v$, respectivamente. Como P_1 e P_2 não contêm o mesmo conjunto de arestas, conseqüentemente eles devem diferir. Se isto acontecer apenas depois de um deles acabar, o resto do outro caminho é um ciclo simples de v a v. Caso contrário, podemos supor que $x_0 = y_0$, $x_1 = y_1$, ..., $x_i = y_i$, mas $x_{i+1} \neq y_{i+1}$. Siga o caminho y_i, y_{i+1}, y_{i+2}, e assim por diante, até que ele encontre novamente um vértice em P_1. Uma vez que estejamos de volta em P_1, siga-o, para frente ou para trás como necessário, de volta a x_i. Como $x_i = y_i$, isto forma um ciclo que deve ser simples, pois nenhuma aresta entre os x_ks pode se repetir e nenhuma aresta entre os x_ks pode ser igual a um dos Y_ls que usamos. **51.** O grafo G é conexo se e somente se todo elemento fora da diagonal de $\mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots + \mathbf{A}^{n-1}$ for positivo, em que $\mathbf{A}$ é a matriz de adjacência de G. **53.** Se o grafo for bipartido, digamos, com partes A e B, então os vértices em todo caminho devem ficar alternadamente em A e B. Portanto, um caminho que comece em A, digamos, acabará em B depois de um número ímpar de passos e em A depois de um número par de passos. Como um ciclo termina no mesmo vértice onde começou, o comprimento deve ser par. Reciprocamente, suponha que todos os ciclos tenham comprimento par; devemos mostrar que o grafo é bipartido. Podemos supor que o grafo seja conexo, pois se não for, podemos simplesmente trabalhar em uma componente de cada vez. Seja v um vértice do grafo, e seja A o conjunto de todos os vértices para os quais exista um caminho de comprimento ímpar começando em v, e seja B o conjunto de todos os vértices para os quais exista um caminho de comprimento par começando em v. Como a componente é conexa, todo vértice está em A ou em B. Nenhum vértice pode estar em ambos, A e B, pois se um estivesse, então seguindo o caminho de comprimento ímpar v para aquele vértice e então de volta ao longo do caminho de comprimento par daquele vértice a v produziria um ciclo ímpar, o que contraria a hipótese. Assim, o conjunto dos vértices foi dividido em dois conjuntos. Para mostrar que toda aresta tem extremidades em partes diferentes, suponha que xy seja uma aresta, em que $x \in A$. Então o caminho de comprimento ímpar de v a x seguido por xy produz um caminho de comprimento par de v a y, de modo que $y \in B$. (Analogamente, se $x \in B$.) **55.** $(H_1W_1H_2W_2\langle\text{barco}\rangle, \emptyset) \rightarrow (H_2W_2, H_1W_1\langle\text{barco}\rangle) \rightarrow (H_1H_2W_2\langle\text{barco}\rangle, W_1) \rightarrow (W_2, H_1W_1H_2\langle\text{barco}\rangle) \rightarrow (H_2W_2\langle\text{barco}\rangle, H_1W_1) \rightarrow (\emptyset, H_1W_1H_2W_2\langle\text{barco}\rangle)$

Seção 9.5

1. Nenhum dos dois. **3.** Nenhum ciclo euleriano; *a*, *e*, *c*, *e*, *b*, *e*, *d*, *b*, *a*, *c*, *d* **5.** *a*, *b*, *c*, *d*, *c*, *e*, *d*, *b*, *e*, *a*, *e*, *a* **7.** *a*, *i*, *h*, *g*, *d*, *e*, *f*, *g*, *c*, *e*, *h*, *d*, *c*, *a*, *b*, *i*, *c*, *b*, *h*, *a* **9.** Não, A ainda tem grau ímpar. **11.** Quando o grafo, no qual os vértices representam interseções e as arestas, ruas, tiver um caminho euleriano. **13.** Sim. **15.** Não. **17.** Se existir um caminho euleriano, conforme o seguimos, cada vértice, exceto os vértices inicial e final, deve ter grau de entrada e grau de saída iguais, pois sempre que chegarmos em um vértice por uma aresta, o deixamos por uma outra aresta. O vértice inicial deve ter grau de saída 1 a mais que o seu grau de entrada, pois usamos uma aresta saindo deste vértice e sempre que o visitarmos de novo, usamos uma aresta levando a ele e uma deixando-o. Analogamente, o vértice final deve ter grau de entrada 1 a mais que seu grau de saída. Como o caminho euleriano com direções removidas produz um caminho entre quaisquer dois vértices, no grafo não orientado subjacente, o grafo é fracamente conexo. Reciprocamente, suponha que o grafo satisfaça as condições sobre os graus do enunciado. Se adicionarmos mais uma aresta a partir do vértice de grau com saída deficiente para o vértice com grau de entrada deficiente, então o grafo tem todo vértice com grau de entrada e grau de saída iguais. Como o grafo ainda é fracamente conexo, pelo Exercício 16 este novo grafo tem um ciclo euleriano. Agora, remova a aresta adicionada para obter um caminho euleriano. **19.** Nenhum dos dois. **21.** Nenhum ciclo euleriano; *a*, *d*, *e*, *d*, *b*, *a*, *e*, *c*, *e*, *b*, *c*, *b*, *e*. **23.** Nenhum dos dois. **25.** Siga o mesmo procedimento do Algoritmo 1, tomando cuidado para seguir as orientações das arestas. **27. a)** $n = 2$. **b)** Nenhum. **c)** Nenhum. **d)** $n = 1$ **29.** Exercício 1:1 vez; Exercícios 2–7: 0 vezes. **31.** *a*, *b*, *c*, *d*, *e*, *a* é um ciclo hamil-

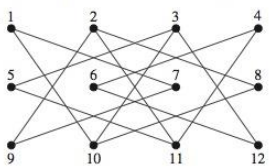
toniano. **33.** Não existe nenhum ciclo hamiltoniano, pois uma vez que um ciclo e que queríamos fosse atingido, ele não teria nenhum outro lugar para ir. **35.** Não existe nenhum ciclo hamiltoniano, pois toda aresta no grafo é incidente a um vértice de grau 2 e, portanto, não pode ser um ciclo. **37.** *a, b, c, f, d, e* é um caminho hamiltoniano. **39.** *f, e, d, a, b, c* é um caminho hamiltoniano. **41.** Não existe nenhum caminho hamiltoniano. Existem oito vértices de grau 2, e apenas dois deles podem ser vértices finais de um caminho. Para cada um dos outros seis, suas duas arestas incidentes não podem ser um caminho. Não é difícil ver que se for para existir um caminho hamiltoniano, exatamente um dos vértices de canto deve ser um final, e isto é impossível. **43.** *a, b, c, f, i, h, g, d, e* é um caminho hamiltoniano. **45.** $m = n \geq 2$ **47. a)** (i) Não, (ii) Não, (iii) Sim **b)** (i) Não, (ii) Não, (iii) Sim. **c)** (i) Sim, (ii) Sim, (iii) Sim **d)** (i) Sim, (ii) Sim, (iii) Sim. **49.** O resultado é trivial para $n = 1$: o código é 0, 1. Suponha que temos um código de Gray de ordem n. Seja $c_1, ..., c_k$, $k = 2^n$ um destes códigos. Então, $0c_1, ..., 0c_k, 1c_k, ..., 1c_1$ é um código de Gray de ordem $n+1$.

51. procedure *Fleury* ($G = (V, E)$: multigrafo conexo com os graus de todos os vértices pares, $V = \{v_1, ..., v_n\}$)

$v := v_1$
ciclo $:= v$
$H := G$
while H tiver arestas
begin
$e :=$ primeira aresta com extremidade v em H (com relação à listagem de V), tal que e não seja uma aresta de corte de H, se existir alguma, e simplesmente a primeira aresta em H com extremidade v, caso contrário
$w :=$ outra extremidade de e
ciclo $:=$ *ciclo* com *e, w* adicionado
$v := w$
$H := H - e$
end {*ciclo* é um ciclo euleriano}

53. Se G tiver um ciclo euleriano, ele também é um caminho euleriano. Se não, adicione uma aresta entre os dois vértices de grau ímpar e aplique o algoritmo para obter um ciclo euleriano. Então, remova a nova aresta. **55.** Suponha que $G = (V, E)$ seja um grafo bipartido com $V = V_1 \cup V_2$, em que nenhuma aresta conecta um vértice em V_1 e um vértice em V_2. Suponha que G tenha um ciclo hamiltoniano. Esse ciclo deve ser da forma $a_1, b_1, a_2, b_2, ..., a_k, b_k, a_1$, em que $a_i \in V_1$ e $b_i \in V_2$ para $i = 1, 2, ..., k$. Como o ciclo hamiltoniano visita cada vértice exatamente uma vez, exceto por v_1, em que ele começa e termina, o número de vértices no grafo é igual a $2k$, um número par. Logo, um grafo bipartido com um número ímpar de vértices não pode ter um ciclo hamiltoniano.

57.



59. Representamos os quadrados de um tabuleiro 3×4 como a seguir:

1	2	3	4
5	6	7	8
9	10	11	12

Um passeio de cavalo pode ser feito pelos seguintes movimentos 8, 10, 1, 7, 9, 2, 11, 5, 3, 12, 6, 4. **61.** Representamos os quadrados de um tabuleiro 4×4 como a seguir:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Existem apenas dois movimentos de cada um dos quatro quadrados de canto. Se incluirmos todas as arestas 1–10, 1–7, 16–10 e 16–7, é completado um ciclo muito cedo, de modo que pelo menos uma destas arestas deve estar faltando. Sem perda de generalidade, suponha que o caminho começa com 1–10, 10–16, 16–7. Agora, os únicos movimentos a partir do quadrado 3 são para os quadrados 5, 10 e 12, e o quadrado 10 já tem duas arestas incidentes. Portanto, 3–5 e 3–12 devem estar no ciclo hamiltoniano. Analogamente, as arestas 8–2 e 8–15 devem estar no ciclo. Agora, os únicos movimentos a partir do quadrado 9 são para os quadrados 2, 7 e 15. Se existirem arestas a partir do quadrado 9 para ambos os quadrados 2 e 15, seria completado um ciclo muito cedo. Assim, a aresta 9–7 deve estar no ciclo dando ao quadrado 7 seu conjunto completo de arestas. Mas agora, o quadrado 14 é forçado a ser ligado aos quadrados 5 e 12, completando um ciclo muito cedo (5–14–12–3–5). Esta contradição mostra que não existe nenhum passeio de cavalo em um tabuleiro 4×4. **63.** Como existem mn quadrados em um tabuleiro $m \times n$, se ambos, m e n, forem ímpares, existe um número ímpar de quadrados. Como, pelo Exercício 62, o grafo correspondente é bipartido, pelo Exercício 55, ele não tem nenhum ciclo hamiltoniano. Logo, não existe nenhum passeio de cavalo reentrante. **65. a)** Se G não tiver um ciclo hamiltoniano, continue tanto quanto possível a adicionar arestas uma de cada vez, de maneira que não obtenhamos um grafo com um ciclo hamiltoniano. Isto não pode continuar para sempre, pois uma vez que tenhamos formado o grafo completo pela adição de todas as arestas que faltam, existe um ciclo hamiltoniano. Sempre que o processo parar, teremos obtido um grafo H (necessariamente não completo) com a propriedade desejada. **b)** Adicione mais uma aresta a H. Isto produz um ciclo hamiltoniano, que usa a aresta adicionada. O caminho que consiste neste ciclo com a aresta adicionada omitida é um caminho hamiltoniano em H. **c)** Claramente, v_1 e v_n não são adjacentes em H, pois H não tem nenhum ciclo hamiltoniano. Portanto, eles não são adjacentes em G. Mas a hipótese era a de que a soma dos graus dos vértices não adjacentes em G era pelo menos n. Esta

desigualdade pode ser reescrita como $n - \text{gr}(v_n) \leq \text{gr}(v_1)$. Mas $n - \text{gr}(v_n)$ é simplesmente o número de vértices não adjacentes a v_n. **d)** Como não existe nenhum vértice depois de v_n no caminho hamiltoniano, v_n não está em S. Cada um dos $\text{gr}(v_1)$ vértices adjacentes a v_1 dão origem a um elemento de S, de modo que S contém $\text{gr}(v_1)$ vértices. **e)** Pela parte (c) existem no máximo $\text{gr}(v_1) - 1$ vértices diferentes de v_n que não são adjacentes a v_n, e pela parte (d) existem $\text{gr}(v_1)$ vértices em S, nenhum dos quais é v_n. Desse modo, pelo menos um vértice de S é adjacente a v_n. Por definição, se v_k for esse vértice, então H contém arestas $v_k v_n$ e $v_1 v_{k+1}$, em que $1 < k < n - 1$. **f)** Agora, $v_1, v_2, ..., v_{k-1}, v_k, v_n, v_{n-1}, ..., v_{k+1}, v_1$ é um ciclo hamiltoniano em H, contradizendo a construção de H. Portanto, nossa hipótese que G não tinha originalmente um ciclo hamiltoniano está errada e nossa demonstração por contradição está completa.

Seção 9.6

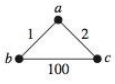
1. a) Os vértices são as paradas, as arestas ligam paradas adjacentes, os pesos são os tempos necessários para se deslocar entre paradas adjacentes. **b)** O mesmo que na parte (a), exceto que os pesos são as distâncias entre as paradas adjacentes. **c)** O mesmo que na parte (a), exceto que os pesos são as tarifas entre as paradas. **3.** 16 **5.** Exercício 2: *a, b, e, d, z*; Exercício 3: *a, c, d, e, g, z*; Exercício 4: *a, b, e, h, l, m, p, s, z* **7. a)** *a, c, d* **b)** *a, c, d, f* **c)** *c, d, f* **e)** *b, d, e, g, z* **9. a)** Orientado **b)** Via Nova York **c)** Via Atlanta e Chicago **d)** Via Nova York **11. a)** Via Chicago **b)** Via Chicago **c)** Via Los Angeles **d)** Via Chicago **13. a)** Via Chicago **b)** Via Chicago **c)** Via Los Angeles **d)** Via Chicago **15.** Não pare o algoritmo quando z for adicionado ao conjunto S. **17. a)** Via Woodbridge, via Woodbridge e Camden **b)** Via Woodbridge, via Woodbridge e Camden **19.** Por exemplo, excursões turísticas, limpeza de ruas.

21.

	a	*b*	*c*	*d*	*e*	*z*
a	4	3	2	8	10	13
b	3	2	1	5	7	10
c	2	1	2	6	8	11
d	8	5	6	4	2	5
e	10	7	8	2	4	3
z	13	10	11	5	3	6

23. $O(n^3)$ **25.** *a-c-b-d-a* (ou o mesmo ciclo começando em algum outro ponto, mas percorrendo os vértices na mesma ordem ou na ordem inversa) **27.** São Francisco–Denver–Detroit–Nova York–Los Angeles–São Francisco (ou o mesmo ciclo começando em algum outro ponto, mas percorrendo os vértices na mesma ordem ou na ordem inversa)

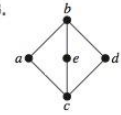
29. Considere este grafo:



O ciclo *a-b-a-c-a* visita cada vértice pelo menos uma vez (e o vértice *a* duas vezes) e tem peso total 6. Todo ciclo hamiltoniano tem peso total 103.

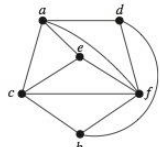
Seção 9.7

1. Sim. **3.**



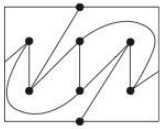
5. Não.

7. Sim.



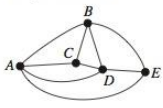
9. Não **11.** É formado um triângulo pela representação planar do subgrafo de K_5, que consiste nas arestas que ligam v_1, v_2 e v_3. O vértice v_4 deve ser colocado ou dentro do triângulo ou fora dele. Consideraremos apenas o caso no qual v_4 está dentro do triângulo; o outro caso é parecido. Desenhando as três arestas de v_1, v_2 e v_3 a v_4 obtemos quatro regiões. Não importa em qual destas regiões v_5 esteja, é possível ligá-lo a apenas três, e não a todos os quatro, dos outros vértices. **13.** 8 **15.** Como não existem laços ou arestas múltiplas e nenhum ciclo de comprimento 3, e o grau da região ilimitada é pelo menos 4, cada região tem grau pelo menos 4. Assim, $2e \geq 4r$, ou $r \leq e/2$. Mas $r = e - v + 2$, de modo que temos $e - v + 2 \leq e/2$, o que implica que $e \leq 2v - 4$. **17.** Como no argumento na demonstração do Corolário 1, temos $2e \geq 5r$ e $r = e - v + 2$. Assim, $e - v + 2 \leq 2e/5$, o que implica que $e \leq (5/3)v - (10/3)$. **19.** Apenas (a) e (c) **21.** Não é homeomorfo a $K_{3,3}$. **23.** Planar **25.** Não planar **27. a)** 1 **b)** 3 **c)** 9 **d)** 2 **e)** 4 **f)** 16 **29.** Desenhe $K_{m,n}$ como descrito na sugestão. O número de cruzamentos é quatro vezes o número no primeiro quadrante. Os vértices no eixo x à direita da origem são (1, 0), (2, 0), ..., $(m/2, 0)$ e os vértices no eixo y acima da origem são (0, 1), (0, 2), ..., $(0, n/2)$. Obtemos todos os cruzamentos escolhendo quaisquer dois números a e b com $1 \leq a < b \leq m/2$ e dois números r e s com $1 \leq r < s \leq n/2$; obtemos exatamente um cruzamento no grafo entre a aresta que conecta $(a, 0)$ e $(0, s)$ e a aresta que conecta $(b, 0)$ e $(0, r)$. Logo, o número de cruzamentos no primeiro quadrante é $C(\frac{m}{2}, 2) \cdot C(\frac{n}{2}, 2) = \frac{(m/2)(m/2-1)}{2} \cdot \frac{(n/2)(n/2-1)}{2}$. Logo, o número total de cruzamentos é $4 \cdot \frac{mn}{2}(m-2)(n-2)/64 = mn(m-2)(n-2)/16$. **31. a)** 2 **b)** 2 **c)** 2 **d)** 2 **e)** 2 **f)** 2 **33.** A fórmula é válida para $n \leq 4$. Se $n > 4$, pelo Exercício 32, a espessura de K_n é pelo menos $C(n, 2)/(3n - 6) = (n + 1 + \frac{2}{n-2})/6$ arredondado para cima. Como essa quantidade nunca é um inteiro, ela é igual ao próximo inteiro maior arredondado para baixo, o qual é igual a $\lfloor (n + 7)/6 \rfloor$. **35.** Isto segue do Exercício 34, pois $K_{m,n}$ tem mn arestas e $m + n$ vértices e não tem nenhum triângulo, já que é bipartido.

37.

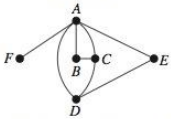


Seção 9.8

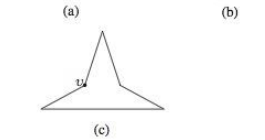
1. Quatro cores



3. Três cores



5. 3 **7.** 3 **9.** 2 **11.** 3 **13.** Grafos sem arestas. **15.** 3, se n for par, 4, se n for ímpar. **17.** Período 1: Math 115, Math 185; período 2: Math 116, CS 473; período 3: Math 195, CS 101; período 4: CS 102; período 5: CS 273 **19.** 5 **21.** Exercício 5: 3 Exercício 6: 6 Exercício 7: 3 Exercício 8: 4 Exercício 9: 3 Exercício 10: 6 Exercício 11: 4 **23.** 5 **25.** Cor 1: *e, f, d*; cor 2: *c, a, i, g*; cor 3: *h, b, j* **27.** Cor C_6 **29.** São necessárias quatro cores para colorir W_n quando n for um inteiro ímpar maior que 1, pois são necessárias três cores para a borda (veja o Exemplo 4), e o vértice central, sendo adjacente a todos os vértices da borda, precisará de uma quarta cor. Para ver que o gráfico obtido de W_n pela remoção de uma aresta pode ser colorido com três cores, considere dois casos. Se removermos uma aresta da borda, então podemos colorir a borda com duas cores, começando em uma extremidade da aresta removida e usando as cores alternadamente em torno da borda restante. A terceira cor é então associada ao vértice central. Se removermos uma aresta do tipo raio, podemos colorir a borda usando a cor #1 para a extremidade na borda da aresta removida e as cores #2 e #3 alternadamente para os vértices restante na borda, e então associar a cor #1 ao centro. **31.** Suponha que G seja cromaticamente k-crítico, mas que tenha um vértice v de grau menor que ou igual a $k-2$. Remova de G uma das arestas incidentes a v. Pela definição de "k-crítico", o grafo resultante pode ser colorido com $k-1$ cores. Agora, reponha a aresta que falta e use essa coloração para todos os vértices, exceto v. Como temos uma coloração própria no grafo menor, nenhum par de vértices adjacentes tem a mesma cor. Além disso, v tem no máximo $k-2$ vizinhos, de modo que podemos colorir v com uma cor não usada para obter uma ($k-1$)-coloração de G. Isso contradiz o fato de G ter número cromático k. Portanto, nossa hipótese estava errada, e todo vértice de G deve ter grau pelo menos $k-1$. **33. a)** 6 **b)** 7 **c)** 9 **d)** 11 **35.** Represente freqüências por cores e zonas por vértices. Ligue dois vértices com uma aresta se as zonas que esses vértices representarem interferirem uma com a outra. Então, uma coloração k-upla é precisamente uma associação de freqüências que evite interferência. **37.** Usamos indução no número de vértices do grafo. Todo grafo com cinco ou menos vértices pode ser colorido com cinco ou menos cores, pois cada vértice pode ficar com uma cor diferente. Isto resolve o(s) caso(s) base. Assim, supomos que todos os grafos com k vértices podem ser 5-coloridos e consideramos um grafo G com $k+1$ vértices. Pelo Corolário 2 da Seção 9.7, G tem um vértice v com grau no máximo 5. Remova v para formar o grafo G'. Como G' tem apenas k vértices, nós o 5-colorimos pela hipótese de indução. Se os vizinhos de v não usarem todas as cinco cores, então podemos 5-colorir G associando a v uma cor que ainda não foi usada por nenhum de seus vizinhos. A dificuldade ocorre se v tiver cinco vizinhos, e cada um tiver uma cor diferente na 5-coloração de G'. Suponha que os vizinhos de v, quando considerados em sentido horário em torno de v, sejam a, b, c, m e p. (Esta ordem é determinada pelo sentido horário das curvas que representam as arestas incidentes a v.) Suponha que as cores dos vizinhos sejam azul-celeste, azul, verde-amarelado, magenta e púrpura, respectivamente. Considere o subgrafo azul-celeste-verde-amarelado (isto é, os vértices em G coloridos de azul-celeste ou verde-amarelado e todas as arestas entre eles). Se a e c não estiverem na mesma componente deste grafo, então na componente que contém a podemos trocar estas duas cores (pinte os vértices azul-celeste de verde-amarelado e vice-versa), e G' ainda estará bem colorido. Isto torna a verde-amarelado, de modo que agora podemos colorir v de azul-celeste, e G foi bem colorido. Se a e c estiverem na mesma componente, então existe um caminho entre os vértices alternadamente coloridos de azul-celeste e verde-amarelado ligando a e c. Este caminho junto com as arestas av e vc divide o plano em duas regiões, com b em uma delas e m na outra. Se agora trocarmos azul e magenta em todos os vértices na mesma região que b, ainda teremos uma boa coloração de G', mas agora o azul está disponível para v. Também neste caso, encontramos uma boa coloração de G. Isto completa o passo de indução e o teorema está demonstrado. **39.** Seguimos a dica. Como a medida dos ângulos interiores de um pentágono totaliza 540°, não pode haver três ângulos interiores que meçam mais de 180° (ângulos reflexos). Se não existir nenhum ângulo reflexo, então o pentágono é convexo, e um guarda colocado em qualquer vértice pode ver todos os pontos. Se houver um ângulo reflexo, então o pentágono deve se parecer essencialmente com a figura (a) abaixo e um guarda no vértice v pode ver todos os pontos. Se houver dois ângulos reflexos, então eles podem ser adjacentes ou não adjacentes (figuras (b) e (c)); em qualquer dos dois casos, um guarda no vértice v pode ver todos os pontos. [Na figura (c), escolha o vértice reflexo mais próximo do lado de baixo.] Assim, para todos os pentágonos, um guarda é suficiente, de modo que $g(5) = 1$.



41. A figura indicada na dica (generalizada para ter k dentes para qualquer $k \geq 1$) tem $3k$ vértices. Os conjuntos de posições das quais as pontas dos diferentes dentes são visíveis são disjuntos. Portanto, é necessário um guarda separado para cada um dos k dentes, de modo que pelo menos k guardas são necessários. Isto mostra que $g(3k) \geq k = \lfloor 3k/3 \rfloor$. Se $n = 3k + i$, em que $0 \leq i \leq 2$, então $g(n) \geq g(3k) \geq k = \lfloor n/3 \rfloor$.

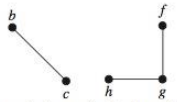
Exercícios Complementares

1. 2500 **3.** Sim **5.** Sim **7.** $\sum_{i=1}^{m} n_i$ vértices, $\sum_{i<j} n_i n_j$ arestas

9. a)

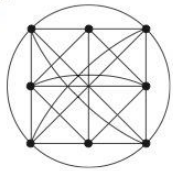
b)

c)

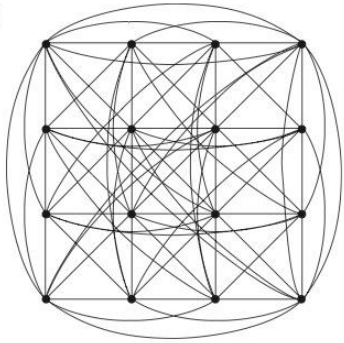


11. Subgrafos completos com o seguinte conjunto de vértices: $\{b, c, e, f\}$, $\{a, b, g\}$, $\{a, d, g\}$, $\{d, e, g\}$, $\{b, e, g\}$ **13.** Subgrafos completos com o seguinte conjunto de vértices: $\{b, c, d, j, k\}$, $\{a, b, j, k\}$, $\{e, f, g, i\}$, $\{a, b, i\}$, $\{a, i, j\}$, $\{b, d, e\}$, $\{b, e, i\}$, $\{b, i, j\}$, $\{g, h, i\}$, $\{h, i, j\}$ **15.** $\{c, d\}$ é um conjunto dominante mínimo.

17. a)



b)



19. a) 1 **b)** 2 **c)** 3 **21. a)** Um caminho de u a v em um grafo G induz um caminho de $f(u)$ a $f(v)$ em um grafo isomorfo H. **b)** Suponha que f seja um isomorfismo de G em H. Se $v_0, v_1, ..., v_n, v_0$ é um ciclo hamiltoniano em G, então $f(v_0), f(v_1), ..., f(v_n), f(v_0)$ deve ser um ciclo hamiltoniano em H, pois ele ainda é um ciclo e $f(v_i) \neq f(v_j)$ para $0 \leq i < j \leq n$. **c)** Suponha que f seja um isomorfismo de G em H. Se $v_0, v_1, ..., v_n, v_0$ é um ciclo euleriano em G, então $f(v_0), f(v_1), ..., f(v_n), f(v_0)$ deve ser um ciclo euleriano em H, pois ele é um ciclo que contém cada aresta exatamente uma vez. **d)** Dois grafos isomorfos devem ter o mesmo número de cruzamentos, pois eles podem ser desenhados exatamente da mesma maneira no plano. **e)** Suponha que f seja um isomorfismo de G a H. Então, v é isolado em G se e somente se $f(v)$ é isolado em H. Logo, os grafos devem ter o mesmo número de vértices isolados. **f)** Suponha que f seja um isomorfismo de G em H. Se G for bipartido, então o conjunto de vértices de G pode ser dividido em V_1 e V_2 sem nenhuma aresta ligando um vértice em V_1 com um em V_2. Então, o conjunto de vértices de H pode ser dividido em $f(V_1)$ e $f(V_2)$ sem nenhuma aresta ligando um vértice em $f(V_1)$ a um em $f(V_2)$. **23.** 3 **25. a)** Sim. **b)** Não. **27.** Não. **29.** Sim. **31.** Se e for uma aresta de corte com extremidades u e v, então se orientarmos e de u a v, não existirá nenhum caminho no grafo orientado de v a u, ou então e não seria uma aresta de corte. Um raciocínio análogo funciona se orientarmos e de v a u. **33.** $n-1$ **35.** Faça com que os vértices representem as galinhas. Incluímos a aresta (u, v) no grafo se e somente se a galinha u dominar a galinha v. **37. a)** 4 **b)** 2 **c)** 3 **d)** 4 **e)** 4 **f)** 2 **39. a)** Suponha que $G = (V, E)$. Sejam $a, b \in V$. Devemos mostrar que a distância entre a e b em $\overline{G}$ é no máximo 2. Se $\{a, b\} \notin E$, esta distância é 1, assim, suponha que $\{a, b\} \in E$. Como o diâmetro de G é maior que 3, existem vértices u e v, tal que a distância em G entre u e v é maior que 3. Ou u ou v, ou ambos, não está no conjunto $\{a, b\}$. Suponha que u seja diferente tanto de a quanto de b. Ou $\{a, u\}$ ou $\{b, u\}$ pertence a E; caso contrário a, u, b seria um caminho em $\overline{G}$ de comprimento 2. Assim, sem perda de generalidade, suponha que $\{a, u\} \in E$. Então, v não pode ser a ou b e, pelo mesmo raciocínio, $\{a, v\} \in E$ ou $\{b, v\} \in E$. Em ambos os casos, isto dá um caminho de comprimento menor que ou igual a 3 de u a v em G, uma contradição. **b)** Suponha que $G = (V, E)$. Sejam $a, b \in V$. Devemos mostrar que a distância entre a e b em $\overline{G}$ não excede 3. Se $\{a, b\} \notin E$, o resultado segue, e assim, suponha que $\{a, b\} \in E$. Como o diâmetro de G é maior que ou igual a 3, existem vértices u e v, tal que a distância em G entre u e v é maior que ou igual a 3. Ou u ou v, ou ambos, não está no conjunto $\{a, b\}$. Suponha que u seja diferente tanto de a quanto de b. Ou $\{a, u\} \in E$ ou $\{b, u\} \in E$; caso contrário, a, u, b é um caminho de comprimento 2 em $\overline{G}$. Assim, sem perda de generalidade suponha que $\{a, u\} \in E$. Então, v é diferente tanto de a quanto de b. Se $\{a, v\} \in E$, então u, a, v é um caminho de comprimento 2 em G, de modo que $\{a, v\} \notin E$ e assim $\{b, v\} \in E$ (caso contrário, existiria um caminho a, v, b de comprimento 2 em $\overline{G}$). Portanto, $\{u, b\} \notin E$; caso contrário, u, b, v é um caminho de comprimento 2 em G. Logo, a, v, u, b é um caminho de comprimento 3 em $\overline{G}$, como desejado. **41.** a, b, e, z **43.** a, c, b, d, e, z **45.** Se G é planar, então como $e \leq 3v - 6$, G tem no máximo 27 arestas. (Se G não for conexo, ele terá ainda menos arestas.) Analogamente, $\overline{G}$ tem no máximo 27 arestas. Mas a união de G e $\overline{G}$ é K_{11}, que tem 55 arestas, e $55 > 27 + 27$. **47.** Suponha que G seja colorido com k cores e tenha número de independência i. Uma vez que cada classe de cor deve ser um conjunto independente, cada classe de cor não tem mais de i elementos. Assim, existem no máximo ki vértices.

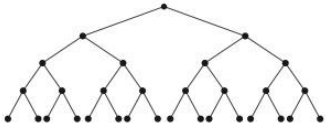
49. **a)** Pelo Teorema 2 da Seção 6.2, a probabilidade de escolher exatamente m arestas é $C(n, m)\ p^m(1-p)^{n-m}$. **b)** Pelo Teorema 2 na Seção 6.4, o valor esperado é np. **c)** Para gerar um grafo rotulado G, à medida que aplicamos o processo a pares de vértices, o número x escolhido arbitrariamente deve ser menor que ou igual a $1/2$ quando G tiver uma aresta entre aquele par de vértices e maior que $1/2$, quando G não tiver uma aresta lá. Logo, a probabilidade de fazer a escolha correta é $1/2$ para cada aresta e $1/2^{C(n,\,2)}$ no total. Desse modo, todos os grafos rotulados são igualmente prováveis. **51.** Suponha que P seja monótona crescente. Se a propriedade de não ter P não fosse mantida sempre que fossem removidas arestas de um grafo simples, existiria um grafo simples G que não tem P e um outro grafo simples G' com os mesmos vértices, mas com algumas das arestas de G que tem P faltando. Mas P é monótona crescente, de modo que como G' tem P, o mesmo ocorre com G obtido pela adição de arestas a G'. Isto é uma contradição. A demonstração da recíproca é análoga.

CAPÍTULO 10

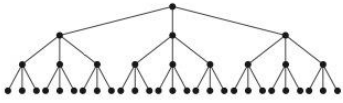
Seção 10.1

1. (a), (c), (e) **3.** **a)** a **b)** $a, b, c, d, f, h, j, q, t$ **c)** $e, g, i, k, l, m, n, o, p, r, s, u$ **d)** q, r **e)** c **f)** p **g)** f, b, a **h)** e, f, l, m, n **5.** Não. **7.** Nível 0: a; nível 1: b, c, d; nível 2: e até k (em ordem alfabética); nível 3: l até r; nível 4: s, t; nível 5: u **9.** **a)** A árvore toda. **b)** c, g, h, o, p e as quatro arestas cg, ch, ho, hp. **c)** apenas e. **11.** **a)** 1 **b)** 2 **13.** **a)** 3 **b)** 9 **15.** A parte "somente se" é o Teorema 2 e a definição de árvore. Suponha que G seja um grafo simples conexo com n vértices e $n-1$ arestas. Se G não for uma árvore, ele contém, pelo Exercício 14, uma aresta cuja remoção produz um grafo G', que ainda é conexo. Se G' não for uma árvore, remova uma aresta para produzir um grafo conexo G''. Repita este procedimento até que o resultado seja uma árvore. Isto exige, no máximo, $n-1$ passos, pois existem apenas $n-1$ arestas. Pelo Teorema 2, o grafo resultante tem $n-1$ arestas, pois ele tem n vértices. Segue que nenhuma aresta foi removida, de modo que G já era uma árvore. **17.** 9999 **19.** 2000 **21.** 999 **23.** 1.000.000 dólares **25.** Não existe tal árvore, pelo Teorema 4, pois isto é impossível para $m = 2$ ou $m = 84$.

27. Árvore binária completa de altura 4:

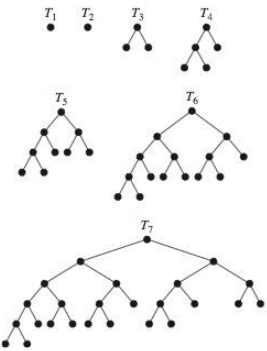


Árvore de grau 3 completa, de altura 3:



29. **a)** Pelo Teorema 3, segue que $n = mi + 1$. Como $i + l = n$, temos $l = n - i$, de modo que $l = (mi + 1) - i = (m-1)i + 1$. **b)** Temos $n = mi + 1$ e $i + l = n$. Assim, $i = n - l$. Segue que $n = m(n - l) + 1$. Isolando n, obtemos $n = (ml-1)/(m-1)$. De $i = n - l$, obtemos $i = [(ml - 1)/(m - 1)] - l = (l - 1)/(m - 1)$. **31.** $n - t$ **33.** **a)** 1 **b)** 3 **c)** 5 **35.** **a)** O diretório pai. **b)** Um subdiretório ou arquivo contido. **c)** Um subdiretório ou arquivo contido no mesmo diretório pai **d)** Todos os diretórios no nome do caminho. **e)** Todos os subdiretórios e arquivos contidos no diretório ou um subdiretório deste diretório, e assim por diante. **f)** O comprimento do caminho para este diretório ou arquivo. **g)** A profundidade do sistema, isto é, o comprimento do caminho mais longo. **37.** Seja $n = 2^k$, em que k é um inteiro positivo. Se $k = 1$, não há nada a demonstrar, pois podemos somar dois números com $n - 1 = 1$ processadores em $\log 2 = 1$ passo. Suponha que possamos somar $n = 2^k$ números em $\log n$ passos usando uma rede conectada por árvore de $n-1$ processadores. Sejam $x_1, x_2, \ldots, x_{2n}$ $2n = 2^{k+1}$ números que queremos somar. A rede conectada em árvore de $2n - 1$ processadores consiste em uma rede conectada em árvore de $n - 1$ processadores junto com dois novos processadores como filhos de cada folha. Em um passo, podemos usar as folhas da rede maior para encontrar $x_1 + x_2, x_3 + x_4, \ldots, x_{2n-1} + x_{2n}$, o que nos dá n números, os quais, pela hipótese de indução, podemos somar em $\log n$ passos usando o resto da rede. Como usamos $\log n + 1$ passos e $\log(2n) = \log 2 + \log n = 1 + \log n$, isto completa a demonstração. **39.** c apenas. **41.** c e h **43.** Suponha que uma árvore T tenha pelo menos dois centros. Sejam u e v centros distintos, ambos com excentricidade e, com u e v não adjacentes. Como T é conexo, existe um caminho simples P de u a v. Seja c um outro vértice qualquer neste caminho. Como a excentricidade de c é pelo menos e, existe um vértice w tal que o único caminho simples de c a w tem comprimento pelo menos e. Claramente, este caminho não pode conter ambos, u e v, ou então existiria um ciclo simples. De fato, este caminho de c a w deixa P e não retorna a P depois de, possivelmente, seguir parte de P em direção a u ou v. Sem perda de generalidade, suponha que este caminho não siga P em direção a u. Então, o caminho de u a c a w é simples e de comprimento maior que e, uma contradição. Assim, u e v são adjacentes. Agora, como quaisquer dois centros são adjacentes, se houvesse mais de dois centros, T conteria K_3, um ciclo simples, como um subgrafo, o que é uma contradição.

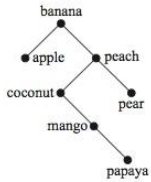
45.



47. A afirmação é de que *toda* árvore com n vértices tem um caminho com comprimento $n - 1$, e foi mostrado apenas que existe uma árvore com n vértices que tem um caminho de comprimento $n - 1$.

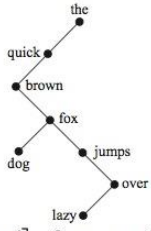
Seção 10.2

1.

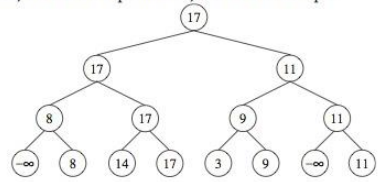


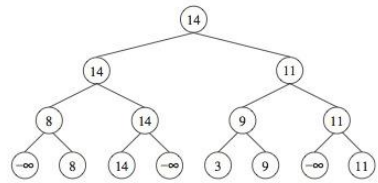
3. a) 3 **b)** 1 **c)** 4 **d)** 5

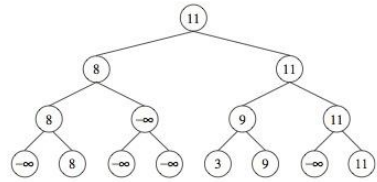
5.

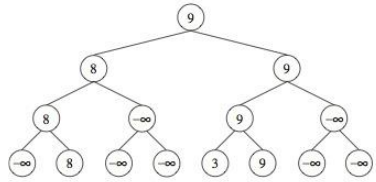


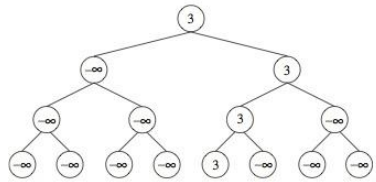
7. Pelo menos $\lceil \log_3 4 \rceil = 2$ pesagens são necessárias, pois existem apenas quatro saídas (pois não é pedido que se determine se a moeda é mais leve ou mais pesada). De fato, duas pesagens seriam suficientes. Comece comparando o peso da moeda 1 com o da moeda 2. Se elas se equilibrarem, compare o peso da moeda 1 com o da moeda 3. Se a moeda 1 e a moeda 3 tiverem o mesmo peso, a moeda 4 é a moeda falsa, e se elas não tiverem o mesmo peso, então a moeda 3 é a moeda falsa. Se a moeda 1 e a moeda 2 não tiverem o mesmo peso, compare novamente o peso da moeda 1 e o da moeda 3. Se elas se equilibrarem, a moeda 2 é a moeda falsa; se elas não se equilibrarem, a moeda 1 é a moeda falsa. **9.** Pelo menos $\lceil \log_3 13 \rceil = 3$ pesagens são necessárias. De fato, três pesagens seriam suficientes. Comece colocando as moedas 1, 2 e 3 no lado esquerdo da balança e as moedas 4, 5 e 6 no lado direito. Se forem iguais, aplique o Exemplo 3 às moedas 1, 2, 7, 8, 9, 10, 11 e 12. Se não forem iguais, aplique o Exemplo 3 a 1, 2, 3, 4, 5, 6, 7 e 8. **11.** O número mínimo é cinco. Chame os elementos de a, b, c e d. Primeiro, compare a e b; então compare c e d. Sem perda de generalidade, suponha que $a < b$ e $c < d$. A seguir, compare a e c. O que for menor, é o menor elemento do conjunto. Novamente sem perda de generalidade, suponha que $a < c$. Finalmente, compare b com ambos, c e d, para determinar completamente a ordem. **13.** Os primeiros dois passos são mostrados no texto. Depois de o 22 ter sido identificado como o segundo maior elemento, substituímos a folha 22 por $-\infty$ na árvore e recalculamos o ganhador no caminho da folha onde 22 costumava estar até a raiz. A seguir, vemos que 17 é o terceiro maior elemento, de modo que repetimos o processo: substitua a folha 17 por $-\infty$ e recalcule. A seguir, vemos que 14 é o quarto maior elemento, de modo que repetimos o processo: substitua a folha 14 por $-\infty$ e recalcule. A seguir, vemos que 11 é o quinto maior elemento, de modo que repetimos o processo: substitua a folha 11 por $-\infty$ e recalcule. O processo continua desta maneira. Determinamos que 9 é o sexto maior elemento, 8 é o sétimo maior elemento e 3, o oitavo maior elemento. As árvores produzidas em todos os passos, exceto o antepenúltimo, são mostradas aqui.







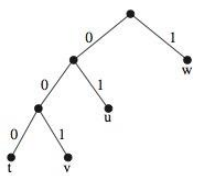




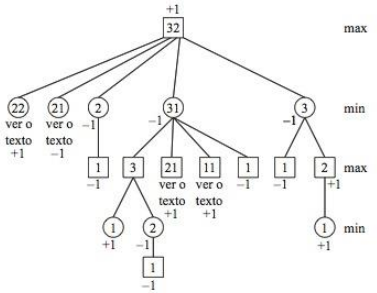
15. O valor de um vértice é o elemento da lista atualmente lá e o rótulo é o nome (isto é, a posição) da folha responsável por aquele valor.

procedure *ordem de torneio*(a_1, ..., a_n)
$k := \lceil \log n \rceil$
construa uma árvore binária de altura k
for $i := 1$ **to** n
  considere o valor da i-ésima folha como sendo a_i e seu rótulo como sendo ele próprio
for $i := n + 1$ **to** 2^k
  considere o valor da i-ésima folha como sendo $-\infty$ e seu rótulo como sendo ele próprio
for $i := k - 1$ **downto** 0
  for cada vértice v no nível i
    considere o valor de v como o maior dos valores de seus filhos e seu rótulo como sendo o rótulo do filho com maior valor
for $i := 1$ **to** n
begin
  $c_i :=$ valor na raiz
  seja v o rótulo da raiz
  considere o valor de v como sendo $-\infty$
  while o rótulo na raiz ainda for v
  begin
    $v := pai(v)$
    considere o valor de v como o maior dos valores de seus filhos e seu rótulo como o rótulo do filho com maior valor
  end
end {c_1, ..., c_n é a lista em ordem não crescente}

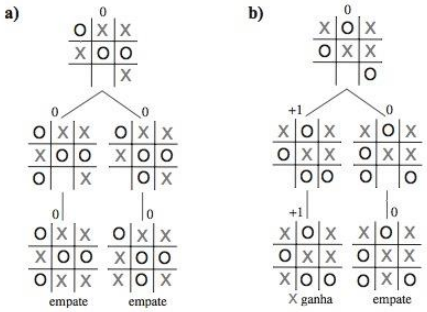
17. $k - 1$, em que $n = 2^k$. **19. a)** Sim. **b)** Não. **c)** Sim. **d)** Sim **21.** *a*: 000, *e*: 001, *i*: 01, *k*: 1100, *o*: 1101, *p*: 11110, *u*: 11111 **23.** a: 11; b: 101; c: 100; d: 01; e: 00; 2,25 bits (Observação: Esta codificação depende dos critérios de desempate, mas o número médio de bits é sempre o mesmo.) **25.** Existem quatro respostas possíveis no total, a mostrada aqui e mais três obtidas desta trocando t e *v* e/ou trocando u e w.

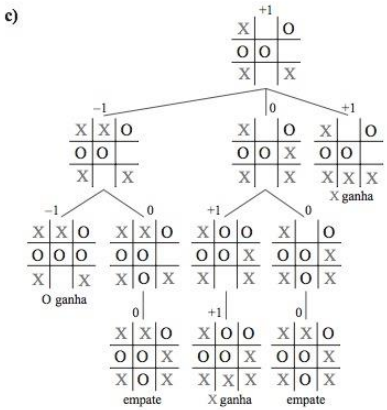


27. A:0001; B:101001; C:11001; D:00000; E:100; F:001100; G:001101; H:0101; I:0100; J:110100101; K:1101000; L:00001; M:10101; N:0110; O:0010; P:101000; Q:1101001000; R:1011; S:0111; T:111; U:00111; V:110101; W:11000; X:11010011; Y:11011; Z:1101001001 **29.** A:2; E:1; N:010; R:011; T:02; Z:00 **31.** n **33.** Como a árvore é bastante grande, indicamos em alguns lugares para "ver o texto". Consulte a Figura 9; a subárvore enraizada nestes vértices quadrados ou circulares é exatamente a mesma que a subárvore correspondente na Figura 9. O primeiro jogador ganha.

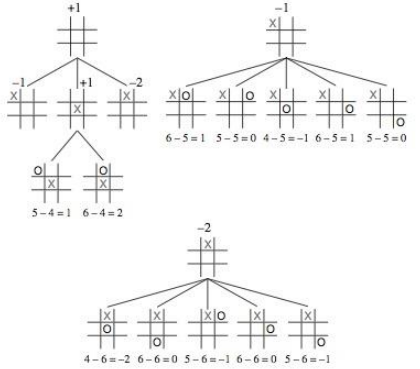


35. a) \$1 **b)** \$3 **c)** −\$3 **37.** Veja as figuras mostradas a seguir. **a)** 0 **b)** 0 **c)** 1 **d)** Esta posição não poderia ter ocorrido em um jogo; esta figura é impossível.



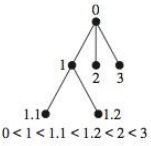


39. Demonstração por indução forte: *Passo base:* Quando existem $n = 2$ pedras em cada pilha, se o primeiro jogador tirar duas pedras de uma pilha, então o segundo jogador tira uma pedra da pilha restante e ganha. Se o primeiro jogador tirar uma pedra de uma pilha, então o segundo jogador tira duas pedras da outra pilha e ganha. *Passo de indução:* Suponha a hipótese de indução de que o segundo jogador sempre pode ganhar se o jogo começar com duas pilhas de j pedras para todo $2 \le j \le k$, em que $k \ge 2$, e considere um jogo com duas pilhas contendo $k + 1$ pedras cada. Se o primeiro jogador tirar todas as pedras de uma das pilhas, então o segundo jogador tira todas, exceto uma pedra da pilha restante e vence. Se o primeiro jogador tirar todas, exceto uma pedra de uma das pilhas, então o segundo jogador tira todas as pedras da outra pilha e vence. Caso contrário, o primeiro jogador deixa j pedras em uma pilha, em que $2 \le j \le k$, e $k + 1$ pedras na outra pilha. O segundo jogador tira o mesmo número de pedras da pilha maior, deixando também j pedras lá. Neste ponto, o jogo consiste em duas pilhas de j pedras cada. Pela hipótese de indução, o segundo jogador no jogo, que também é o segundo jogador em nosso jogo real, pode vencer, e a demonstração por indução forte está completa. **41.** 7; 49 **43.** O valor da árvore é 1. Observação: As segunda e terceira árvores são as subárvores dos dois filhos da raiz na primeira árvore cujas subárvores não estão mostradas por causa de limitação do espaço. Elas devem ser consideradas como unidas à primeira figura.

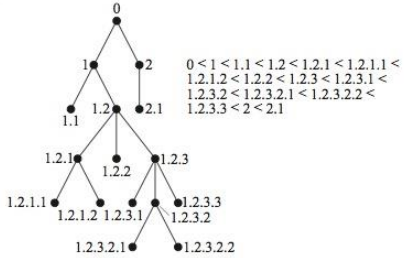


Seção 10.3

1.

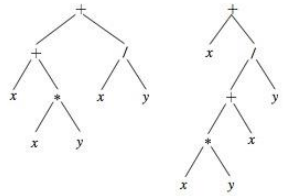


3.



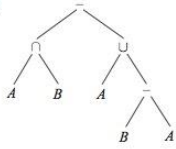
5. Não. **7.** *a, b, d, e, f, g, c* **9.** *a, b, e, k, l, m, f, g, n, r, s, c, d, h, o, i, j, p, q* **11.** *d, b, i, e, m, j, n, o, a, f, c, g, k, h, p, l* **13.** *d, f, g, e, b, c, a* **15.** *k, l, m, e, f, r, s, n, g, b, c, o, h, i, p, q, j, d, a*

17. a)



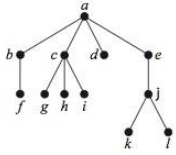
b) $++x*xy/xy, +x/+*xyxy$ **c)** $xxy*+xy/+, xxy*x+y/+$ **d)** $((x+(x*y))+(x/y)), (x+(((x*y)+x)/y))$

19. a)



b) $-\cap A\,B \cup A - B\,A$ **c)** $AB \cap A\,B\,A - \cup -$ **d)** $((A \cap B) - (A \cup (B - A)))$ **21.** 14 **23. a)** 1 **b)** 1 **c)** 4 **d)** 2205

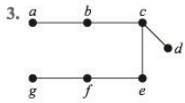
25.



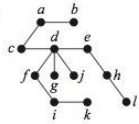
27. Use indução matemática. O resultado é trivial para uma lista com um elemento. Suponha que o resultado seja verda-

deiro para uma lista com n elementos. Para o passo de indução, comece no final. Encontre a seqüência de vértices no final da lista começando com a última folha, terminando com a raiz, cada vértice sendo o último filho do que vem a seguir. Remova esta folha e aplique a hipótese de indução. **29.** c, d, b, f, g, h, e, a em cada caso. **31.** Demonstração por indução matemática. Sejam $S(X)$ e $O(X)$ o número de símbolos e o número de operadores em uma fórmula bem formada X, respectivamente. A afirmação é verdadeira para fórmulas bem formadas de comprimento 1, pois elas têm 1 símbolo e 0 operadores. Suponha que a afirmação seja verdadeira para todas as fórmulas bem formadas de comprimento menor que n. Uma fórmula bem formada de comprimento n deve ser da forma $*XY$, em que $*$ é um operador e X e Y são fórmulas bem formadas de comprimento menor que n. Então, pela hipótese de indução, $S(*XY) = S(X) + S(Y) = [O(X) + 1] + [O(Y) + 1] = O(X) + O(Y) + 2$. Como $O(*XY) = 1 + O(X) + O(Y)$, segue que $S(*XY) = O(*XY) + 1$. **33.** $xy+zx\circ+x\circ$, $xyz++yx++$, $xyxy\circ\circ xy\circ\circ z\circ+$, $xz\times zz+\circ$, $yyyy\circ\circ\circ$, $zx+yz+\circ$, por exemplo.

3.

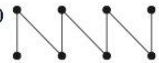


5.



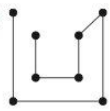
7. a)

b)



c)

d)



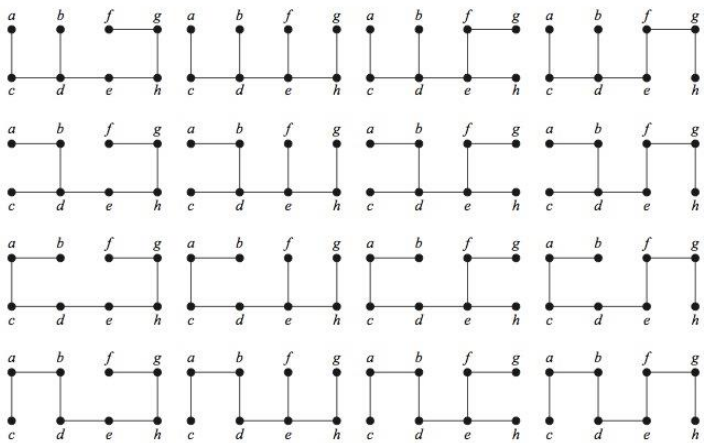
e)

f)

Seção 10.4

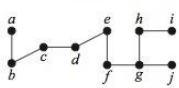
1. $m - n + 1$

9.

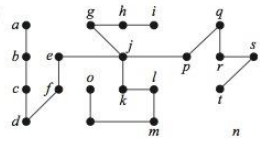


11. a) 3 **b)** 16 **c)** 4 **d)** 5

13.

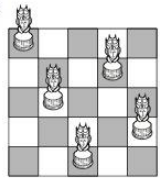


15.

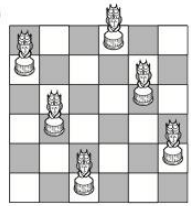


17. **a)** Um caminho de comprimento 6. **b)** Um caminho de comprimento 5. **c)** Um caminho de comprimento 6. **d)** Depende da ordem escolhida para visitar os vértices; pode ser um caminho de comprimento 7. **19.** Com busca em largura, o vértice inicial é o vértice do meio, e os n raios são adicionados à árvore quando este vértice é processado. Portanto, a árvore resultante é $K_{1,n}$. Com busca em profundidade, começamos com o vértice no meio da roda e visitamos um vizinho—um dos vértices da borda. De lá, nos movemos para um vértice adjacente na borda, e assim por diante por toda a volta até termos atingido todos os vértices. Portanto, a árvore geradora resultante é um caminho de comprimento n. **21.** Com a busca em largura, nós nos espalhamos de um vértice de grau m para todos os vértices de grau n como primeiro passo. A seguir, um vértice de grau n é processado, e as arestas dele para todos os vértices restantes de grau m são adicionadas. O resultado é um $K_{1,n-1}$ e um $K_{1,m-1}$ com seus centros ligados por uma aresta. Com a busca em profundidade, nos movemos para frente e para trás de um conjunto da partição para o outro até que não possamos mais prosseguir. Se $m = n$ ou $m = n - 1$, então obtemos um caminho de comprimento $m + n - 1$. Caso contrário, o caminho acaba enquanto alguns vértices no conjunto maior da partição ainda não foram visitados, de modo que voltamos uma ligação no caminho para um vértice v e então visitamos sucessivamente os vértices restantes neste conjunto de v. O resultado é um caminho com arestas extras pendentes saindo de uma extremidade do caminho. **23.** Um conjunto possível de vôos a serem interrompidos é: Boston–Nova York, Detroit–Boston, Boston–Washington, Nova York–Washington, Nova York–Chicago, Atlanta–Washington, Atlanta–Dallas, Atlanta–Los Angeles, Atlanta–St. Louis, St. Louis–Dallas, St. Louis– Detroit, St. Louis–Denver, Dallas–San Diego, Dallas–Los Angeles, Dallas–São Francisco, San Diego–Los Angeles, Los Angeles–São Francisco, São Francisco–Seattle. **25.** Árvores. **27.** Demonstração por indução no comprimento do caminho: Se o caminho tiver comprimento 0, então o resultado é trivial. Se o comprimento for 1, então u é adjacente a v, de modo que u está no nível 1 na árvore geradora em largura. Suponha que o resultado seja verdadeiro para caminhos de comprimento l. Se o comprimento de um caminho for $l + 1$, seja u' o penúltimo vértice em um caminho mais curto de v a u. Pela hipótese de indução, u' está no nível l na árvore geradora em largura. Se u estivesse em um nível que não exceda l, então, claramente, o comprimento do caminho mais curto de v a u também não excederia l. Assim, u ainda não foi adicionado à árvore geradora em largura depois de os vértices do nível l terem sido adicionados. Como u é adjacente a u', ele seria adicionado no nível $l + 1$ (embora a aresta ligando u' e u não seja necessariamente adicionada). **29.** **a)** Nenhuma solução.

b)



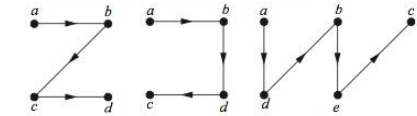
c)



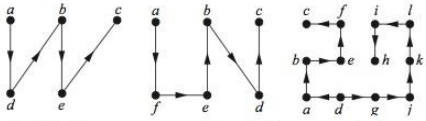
31. Comece em um vértice e prossiga ao longo de um caminho sem repetir vértices enquanto for possível, permitindo o retorno ao começo depois de todos os vértices terem sido visitados. Quando for impossível continuar ao longo de um caminho, retorne por ele e tente uma outra extensão do caminho atual. **33.** Considere a união das árvores geradoras das componentes conexas de G. Elas são disjuntas, de modo que o resultado é uma floresta. **35.** $m - n + c$ **37.** Use busca em profundidade em cada componente. **39.** Se uma aresta uv não for seguida enquanto estivermos processando o vértice u durante o processo de busca em profundidade, então deve ocorrer que o vértice v já foi visitado. Existem dois casos. Se o vértice v tiver sido visitado depois de começarmos a processar u, então, como ainda não terminamos de processar u, v deve aparecer na subárvore enraizada em u (e, portanto, deve ser um descendente de u). Por outro lado, se o processamento de v já tivesse começado antes de começarmos a processar u, então por que esta aresta não foi percorrida naquele momento? Deve ocorrer que ainda não tínhamos terminado de processar v, em outras palavras, que ainda estamos formando a subárvore enraizada em v, de modo que u é um descendente de v, e, portanto, v é um ancestral de u. **41.** Certamente, estes procedimentos produzem árvores geradoras idênticas se o grafo com o qual estivermos trabalhando for ele próprio uma árvore, pois neste caso existe apenas uma árvore geradora (o grafo todo). Entretanto, este é o único caso em que isto ocorre. Se o grafo original tiver outras arestas, então pelo Exercício 39 elas devem ser arestas de retorno e, portanto, ligam um vértice a um ancestral ou a um descendente, enquanto pelo Exercício 40, elas devem conectar vértices no mesmo nível ou em níveis que diferem por 1. Claramente, estas duas possibilidades são mutuamente exclusivas. Portanto, não podem existir arestas além das arestas da árvore se as duas árvores geradoras forem as mesmas. **43.** Como as arestas que não estão na árvore geradora não são seguidas no processo, podemos ignorá-las. Portanto, podemos supor que o grafo era inicialmente uma árvore enraizada. O passo base é trivial (existe apenas um vértice), de modo que supomos a hipótese de indução de que a busca em largura aplicada a árvores com n vértices visita seus vértices na ordem de seu nível na árvore e consideramos a árvore T com $n + 1$ vértices. O último vértice a ser visitado durante a busca em largura desta árvore, digamos v, é aquele que foi adicionado por último à lista dos vértices esperando para serem processados. Ele foi adicionado quando seu pai, digamos u, estava sendo processado. Devemos mostrar que v está no nível mais baixo (isto é, numericamente maior) da árvore. Suponha que não; digamos que o vértice x, cujo pai é o vértice w, está em um nível mais baixo.

Então, w está em um nível mais baixo que u. Claramente, v deve ser uma folha, pois qualquer filho de v não poderia ter sido visto antes de v ser visto. Considere a árvore T' obtida de T pela remoção de v. Pela hipótese de indução, os vértices em T' devem ser processados na ordem de seu nível em T' (que é o mesmo que seu nível em T, e a ausência de v em T' não tem nenhum efeito no resto do algoritmo). Portanto, u deve ter sido processado antes de w, e assim v teria sido colocado na lista de espera antes de x ser, uma contradição. Portanto, v está no nível mais baixo da árvore, e a demonstração está completa. **45.** Modificamos o pseudocódigo dado no Algoritmo 2 inicializando m com 0 no início do algoritmo e adicionando os comandos "$m := m + 1$" e "atribua m ao vértice v" depois do comando que remove o vértice v de L. **47.** Se uma aresta orientada uv não for percorrida enquanto estivermos processando seu vértice final u durante o processo de busca em profundidade, então deve ocorrer que seu vértice inicial v já foi visitado. Existem três casos. Se o vértice v foi visitado depois de começarmos a processar u, então, como ainda não terminamos de processar u, v deve aparecer na subárvore enraizada em u (e, portanto, deve ser um descendente de u), de modo que temos uma aresta progressiva. Caso contrário, o processamento de v já deveria ter começado antes de começarmos a processar u. Se ainda não tivesse acabado (isto é, ainda estamos formando a subárvore enraizada em v), então u é um descendente de v, e, portanto, v é um ancestral de u (temos uma aresta de retorno). Finalmente, se o processamento de v já tivesse terminado, então, por definição, temos uma aresta cruzada. **49.** Seja T a árvore geradora construída na Figura 3 e T_1, T_2, T_3 e T_4 as árvores geradoras na Figura 4. Indique por $d(T', T'')$ a distância entre T' e T''. Então, $d(T, T_1) = 6$, $d(T, T_2) = 4$, $d(T, T_3) = 4$, $d(T, T_4) = 2$, $d(T_1, T_2) = 4$, $d(T_1, T_3) = 4$, $d(T_1, T_4) = 6$, $d(T_2, T_3) = 4$, $d(T_2, T_4) = 2$ e $d(T_3, T_4) = 4$. **51.** Suponha que $e_1 = \{u, v\}$ seja como especificado. Então, $T_2 \cup \{e_1\}$ contém um ciclo simples C que contém e_1. O grafo $T_1 - \{e_1\}$ tem duas componentes conexas; as extremidades de e_1 estão em componentes diferentes. Percorra C a partir de u no sentido oposto a e_1 até chegar no primeiro vértice na mesma componente de v. A aresta que acabamos de cruzar é e_2. Claramente, $T_2 \cup \{e_1\} - \{e_2\}$ é uma árvore, pois e_2 estava em C. Além disso, $T_1 - \{e_1\} \cup \{e_2\}$ é uma árvore, pois e_2 reuniu as duas componentes.

53. Exercício 18: Exercício 19: Exercício 20:



Exercício 21: Exercício 22: Exercício 23:

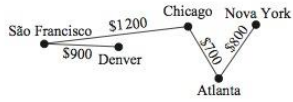


55. Primeiro, construa um ciclo euleriano no grafo orientado. Então, remova deste ciclo toda aresta que vá para um vértice visitado anteriormente. **57.** De acordo com o Exercício 56, um grafo orientado contém um ciclo se e somente se existir alguma aresta de retorno. Podemos detectar arestas de retorno da seguinte maneira. Adicione uma marca em cada vértice v para indicar qual é seu status: ainda não foi visto (a situação inicial), já visto (isto é, posto em T) mas ainda não terminado (isto é, *visit*(v) ainda não terminou) ou terminado (isto é, *visit*(v) já terminou). Algumas linhas extras no Algoritmo 1 efetuarão estes registros. Então, para determinar se um grafo orientado tem um ciclo, temos apenas de verificar quando olharmos para a aresta uv se o status de v é "já visto". Se isto ocorrer alguma vez, então sabemos que existe um ciclo; caso contrário, então não existe nenhum ciclo.

Seção 10.5

1. Deep Springs–Oasis, Oasis–Dyer, Oasis–Silverpeak, Silverpeak–Goldfield, Lida–Gold Point, Gold Point–Beatty, Lida–Goldfield, Goldfield–Tonopah, Tonopah–Manhattan, Tonopah–Warm Springs. **3.** $\{e, f\}$, $\{c, f\}$, $\{e, h\}$, $\{h, i\}$, $\{b, c\}$, $\{b, d\}$, $\{a, d\}$, $\{g, h\}$

5.

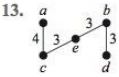


7. $\{e, f\}$, $\{a, d\}$, $\{h, i\}$, $\{b, d\}$, $\{c, f\}$, $\{e, h\}$, $\{b, c\}$, $\{g, h\}$

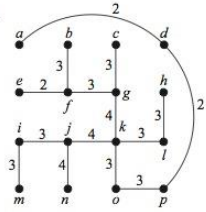
9.

11. Em vez de escolher as arestas de peso mínimo em cada estágio, escolha as arestas de peso máximo em cada estágio, com as mesmas propriedades.

13.

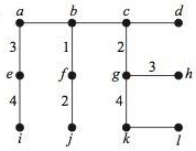


15.



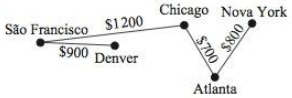
17. Primeiro, encontre uma árvore geradora mínima T do grafo G com n arestas. Então, para $i = 1$ a $n - 1$, remova apenas a i-ésima aresta de T de G e encontre uma árvore geradora mínima para o resto do grafo. Escolha uma destas $n - 1$ árvores com o comprimento mais curto. **19.** Se todas as arestas tiverem pesos diferentes, então, obtém-se uma contradição na demonstração de que o algoritmo de Prim funciona quando uma aresta e_{k+1} é adicionada a T e uma aresta e é removida, em vez de possivelmente produzir uma outra árvore geradora.

21.

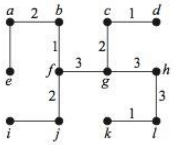


23. O mesmo que o algoritmo de Kruskal, exceto que começa com $T :=$ este conjunto de arestas e itere de $i = 1$ até $i = n - 1 - s$, em que s é o número de arestas com que você começou.

25. a)



b)

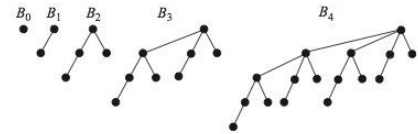


27. Pelo Exercício 24, em cada estágio do algoritmo de Sollin resulta uma floresta. Assim, depois de $n - 1$ arestas terem sido escolhidas, resulta uma árvore. Resta mostrar que esta árvore é uma árvore geradora mínima. Seja T uma árvore geradora mínima com tantas arestas em comum com a árvore de Sollin S quanto possível. Se $T \neq S$, então existe uma aresta $e \in S - T$ adicionada em algum estágio do algoritmo, em que antes deste estágio todas as arestas em S também estão em T. $T \cup \{e\}$ contém um único ciclo simples. Encontre uma aresta $e' \in S - T$ e uma aresta $e'' \in T - S$ neste ciclo e "adjacentes" quando considerando as árvores neste estágio como "super-vértices." Então, pelo algoritmo, $w(e') \leq w(e'')$. Assim, substitua T por $T - \{e''\} \cup \{e'\}$ para produzir uma árvore geradora mínima mais próxima de S do que T estava. **29.** Cada uma das r árvores está ligada a uma outra árvore por uma nova aresta. Assim, existem no máximo $r/2$ árvores no resultado (cada nova árvore contém duas ou mais das antigas árvores). Para fazer isto, precisamos adicionar $r - (r/2) = r/2$ arestas. Como o número de arestas adicionadas é inteiro, ele é pelo menos $\lceil r/2 \rceil$. **31.** Se $k \geq \log n$, então $n/2^k \leq 1$, de modo que $\lceil n/2^k \rceil = 1$, e assim, pelo Exercício 30, o algoritmo termina depois de, no máximo, $\log n$ iterações.

Exercícios Complementares

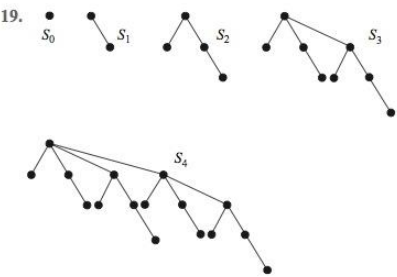
1. Suponha que T seja uma árvore. Então, claramente T não tem nenhum ciclo simples. Se adicionarmos uma aresta e ligando dois vértices não adjacentes, u e v, então obviamente foi formado um ciclo simples, pois, quando e é adicionado a T, o grafo resultante tem arestas demais para ser uma árvore. O único ciclo simples formado é composto pela aresta e junto com o único caminho P em T de v a u. Suponha que T satisfaça as condições dadas. Tudo o que é preciso é mostrar que T é conexo, pois não há nenhum ciclo simples no grafo. Suponha que T não seja conexo. Então, sejam u e v vértices em componentes conexas separadas. Adicionar $e = \{u, v\}$ não satisfaz as condições. **3.** Suponha que uma árvore T tenha n vértices de graus $d_1, d_2, ..., d_n$, respectivamente. Como $2e = \sum_{i=1}^{n} d_i$ e $e = n - 1$, temos $2(n - 1) = \sum_{i=1}^{n} d_i$. Como cada $d_i \geq 1$, segue que $2(n - 1) = n + \sum_{i=1}^{n}(d_i - 1)$, ou que $n - 2 = \sum_{i=1}^{n}(d_i - 1)$. Assim, no máximo $n - 2$ dos termos desta soma podem ser 1 ou mais. Assim, pelo menos dois deles são 0. Segue que $d_i = 1$ para pelo menos dois valores de i. **5.** $2n - 2$. **7.** T não tem nenhum ciclo, de modo que ele não pode ter um subgrafo homeomorfo a $K_{3,3}$ ou K_5. **9.** Pinte cada componente conexa separadamente. Para cada uma destas componentes conexas, primeiro escolha a raiz da árvore, então pinte todos os vértices em níveis pares de vermelho e todos os vértices em níveis ímpares de azul. **11.** Limitante superior: k^h; limitante inferior: $2\lceil k/2 \rceil^{h-1}$.

13.



15. Como B_{k+1} é formado por duas cópias de B_k uma abaixada um nível, a altura aumenta de 1 quando k aumenta de 1. Como B_0 tem altura 0, segue por indução que B_k tem altura k. **17.** Como a raiz de B_{k+1} é a raiz de B_k com um filho adicional (a saber, a raiz do outro B_k), o grau da raiz aumenta de 1 quando k aumenta de 1. Como B_0 tem uma raiz com grau 0, segue por indução que B_k tem uma raiz com grau k.

19.

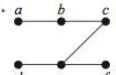


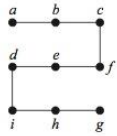
21. Use indução matemática. O resultado é trivial para $k = 0$. Suponha que ele seja verdadeiro para $k - 1$. T_{k-1} é a árvore pai para T. Por indução, a árvore filha para T pode ser obtida de $T_0, ..., T_{k-2}$ da forma descrita. A conexão final de r_{k-2} a r_{k-1} é como descrito na definição de uma árvore S_k.

23. **procedure** *nível*(T: árvore enraizada ordenada com raiz r)
fila := seqüência que consiste apenas na raiz r
while *fila* contiver pelo menos um termo

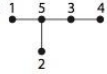
begin
$v :=$ primeiro vértice na fila
liste v
remova v da fila e ponha os filhos de v no final da fila
end

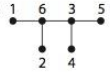
25. Construa a árvore pela inserção de uma raiz para o endereço 0, e então pela inserção de uma subárvore para cada vértice rotulado i, para i um inteiro positivo, construída a partir de subárvores para cada vértice rotulado $i \cdot j$ para j um inteiro positivo, e assim por diante. **27. a)** Sim. **b)** Não. **c)** Sim. **29.** O grafo resultante não tem nenhuma aresta que esteja em mais de um ciclo simples do tipo descrito. Assim, ele é um cactus.

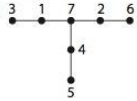
31.


33.


35. a)

b)


c)


d)


37. 6 **39. a)** 0 para 00, 11 para 01, 100 para 10, 101 para 11 (a codificação exata depende de quantas vezes ocorreram desempates, mas todas as versões são equivalentes); $0{,}645n$ para uma seqüência de comprimento n. **b)** 0 para 000, 100 para 001, 101 para 010, 110 para 100, 11100 para 011, 11101 para 101, 11110 para 110, 11111 para 111 (a codificação exata depende de quantas vezes ocorreram desempates, mas todas as versões são equivalentes); $0{,}5326n$ para uma seqüência de comprimento n. **41.** Seja G' o grafo obtido pela remoção de G do vértice v e de todas as arestas incidentes a v. Pode ser obtida uma árvore geradora mínima de G considerando uma aresta de peso mínimo incidente a v junto com uma árvore geradora mínima de G'. **43.** Suponha que a aresta e seja uma aresta de peso mínimo incidente a um vértice v e suponha que T seja uma árvore geradora que não inclui e. Adicione e a T e remova do ciclo simples assim formado a outra aresta do ciclo que contém v. O resultado será uma árvore geradora de peso estritamente menor (pois a aresta removida tem peso maior que o peso de e). Isto é uma contradição, de modo que T deve incluir e.

CAPÍTULO 11

Seção 11.1

1. a) 1 **b)** 1 **c)** 0 **d)** 0 **3. a)** $(1 \cdot 1) + (\overline{0 \cdot 1} + 0) = 1 + (\bar{0} + 0) = 1 + (1 + 0) = 1 + 1 = 1$
b) $(\mathbf{T} \wedge \mathbf{T}) \vee (\neg(\mathbf{F} \wedge \mathbf{T}) \vee \mathbf{F}) \equiv \mathbf{T}$

5. a)

x	y	z	$\bar{x}\bar{y}$
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	1
0	0	1	0
0	0	0	0

b)

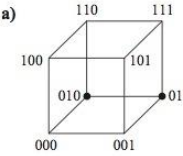
x	y	z	$x + yz$
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

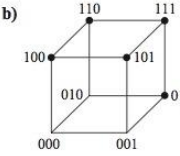
c)

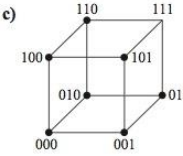
x	y	z	$x\bar{y} + \bar{x}\bar{y}\bar{z}$
1	1	1	0
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	1

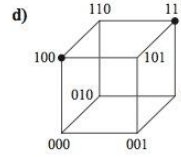
d)

x	y	z	$x(yz + \bar{y}\bar{z})$
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

7. a)


b)


c)


d)


9. (0, 0) e (1, 1) **11.** $x + xy = x \cdot 1 + xy = x(1 + y) = x(y + 1) = x \cdot 1 = x$

13.

x	y	z	$x\bar{y}$	$y\bar{z}$	$\bar{x}z$	$x\bar{y} + y\bar{z} + \bar{x}z$	$\bar{x}\bar{y}$	$\bar{y}z$	$x\bar{z}$	$\bar{x}\bar{y} + \bar{y}z + x\bar{z}$
1	1	1	0	0	0	0	0	0	0	0
1	1	0	0	1	0	1	0	0	1	1
1	0	1	1	0	0	1	0	1	0	1
1	0	0	1	0	0	1	0	0	1	1
0	1	1	0	0	1	1	1	0	0	1
0	1	0	0	1	0	1	1	0	0	1
0	0	1	0	0	1	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0

15.

x	$x+x$	$x \cdot x$
0	0	0
1	1	1

17.

x	$x+1$	$x \cdot 0$
0	1	0
1	1	0

19.

x	y	z	$y+z$	$x+(y+z)$	$x+y$	$(x+y)+z$	yz	$x(yz)$	xy	$(xy)z$
1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	0	0	1	0
1	0	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0
0	1	0	1	1	1	1	0	0	0	0
0	0	1	1	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

21.

x	y	xy	$\overline{(xy)}$	$\bar{x}$	$\bar{y}$	$\bar{x}+\bar{y}$	$x+y$	$\overline{(x+y)}$	$\bar{x}\bar{y}$
1	1	1	0	0	0	0	1	0	0
1	0	0	1	0	1	1	1	0	0
0	1	0	1	1	0	1	1	0	0
0	0	0	1	1	1	1	0	1	1

23. $0 \cdot \bar{0} = 0 \cdot 1 = 0$; $1 \cdot \bar{1} = 1 \cdot 0 = 0$

25.

x	y	$x \oplus y$	$x+y$	xy	$\overline{(xy)}$	$(x+y)\,\overline{(xy)}$	$x\bar{y}$	$\bar{x}y$	$x\bar{y}+\bar{x}y$
1	1	0	1	1	0	0	0	0	0
1	0	1	1	0	1	1	1	0	1
0	1	1	1	0	1	1	0	1	1
0	0	0	0	0	1	0	0	0	0

27. **a)** Verdadeira, como uma tabela de valores pode mostrar. **b)** Falsa; considere $x=1, y=1, z=1$, por exemplo. **c)** Falsa; considere $x=1$, $y=1$, $z=0$, por exemplo. **29.** Pelas leis de De Morgan, o complemento de uma expressão é como o dual, exceto que foi tomado o complemento de cada variável. **31.** 16 **33.** Se substituirmos cada 0 por **F**, 1 por **T**, a soma booleana por $\vee$, o produto booleano por $\wedge$ e $-$ por $\neg$ (e x por p e y por q, de modo que as variáveis parecem representar proposições, e o sinal de igual pelo símbolo de equivalência lógica), então $\overline{xy} = \bar{x} + \bar{y}$ se torna $\neg(p \wedge q) \equiv \neg p \vee \neg q$ e $\overline{x+y} = \bar{x}\bar{y}$ se torna $\neg(p \vee q) \equiv \neg p \wedge \neg q$. **35.** Pelas propriedades de dominação, distributivas e dos elementos neutros, $x \vee x = (x \vee x) \wedge 1 = (x \vee x) \wedge (x \vee \bar{x}) = x \vee (x \wedge \bar{x}) = x \vee 0 = x$. Analogamente, $x \wedge x = (x \wedge x) \vee 0 = (x \wedge x) \vee (x \wedge \bar{x}) = x \wedge (x \vee \bar{x}) = x \wedge 1 = x$. **37.** Como $0 \vee 1 = 1$ e $0 \wedge 1 = 0$ pelas propriedades do elemento neutro e comutativa, segue que $\bar{0} = 1$. Analogamente, como $1 \vee 0 = 1$ e $1 \wedge 0 = 1$, segue que $\bar{1} = 0$. **39.** Primeiro, observe que $x \wedge 0 = 0$ e $x \vee 1 = 1$ para todo x, como pode ser facilmente demonstrado. Para demonstrar a primeira identidade, é suficiente mostrar que $(x \vee y) \vee (\bar{x} \wedge \bar{y}) = 1$ e $(x \vee y) \wedge (\bar{x} \wedge \bar{y}) = 0$. Pelas propriedades associativa, comutativa, distributiva, de dominação e do elemento neutro, $(x \vee y) \vee (\bar{x} \wedge \bar{y}) = y \vee [x \vee (\bar{x} \wedge \bar{y})] = y \vee [(x \vee \bar{x}) \wedge (x \vee \bar{y})] = y \vee [1 \wedge (x \vee \bar{y})] = y \vee (x \vee \bar{y}) = (y \vee \bar{y}) \vee x = 1 \vee x = 1$ e $(x \vee y) \wedge (\bar{x} \wedge \bar{y}) = \bar{y} \wedge [\bar{x} \wedge (x \vee y)] = \bar{y} \wedge [(\bar{x} \wedge x) \vee (\bar{x} \wedge y)] = \bar{y} \wedge [0 \vee (\bar{x} \wedge y)] = \bar{y} \wedge (\bar{x} \wedge y) = \bar{x} \wedge (y \wedge \bar{y}) = \bar{x} \wedge 0 = 0$. A segunda identidade é demonstrada de maneira análoga. **41.** Usando a hipótese, o Exercício 35 e a propriedade distributiva, segue que $x = x \vee 0 = x \vee (x \vee y) = (x \vee x) \vee y = x \vee y = 0$. Analogamente, $y = 0$. Para demonstrar a segunda afirmação, observe que $x = x \wedge 1 = x \wedge (x \wedge y) = (x \wedge x) \wedge y = x \wedge y = 1$. Analogamente, $y = 1$. **43.** Use os exercícios 39 e 41 dos Exercícios Complementares do Capítulo 8 e a definição de um reticulado distributivo complementado para estabelecer os cinco pares de propriedades na definição.

Seção 11.2

1. **a)** $\bar{x}\bar{y}z$ **b)** $\bar{x}y\bar{z}$ **c)** $\bar{x}yz$ **d)** $\overline{xyz}$ **3.** **a)** $xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}y\bar{z} + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$ **b)** $xyz + xy\bar{z} + \bar{x}yz$ **c)** $xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z}$ **d)** $x\bar{y}z + x\bar{y}\bar{z}$ **5.** $wxy\bar{z} + wx\bar{y}z + w\bar{x}yz + \bar{w}xyz + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}yz + \bar{w}x\bar{y}z + \bar{w}\bar{x}y\bar{z}$ **7.** **a)** $\bar{x} + \bar{y} + z$ **b)** $x + y + z$ **c)** $x + \bar{y} + z$ **9.** $y_1 + y_2 + \cdots + y_n = 0$ se e somente se $y_i = 0$ para $i = 1, 2, \ldots, n$. Isto vale se e somente se $x_i = 0$ quando $y_i = x_i$ e $x_i = 1$ quando $y_i = \bar{x}_i$. **11.** **a)** $x + y + z$ **b)** $(x+y+z)(x+y+\bar{z})(x+\bar{y}+z)(\bar{x}+y+z)(\bar{x}+y+\bar{z})$ **c)** $(x+y+z)(x+y+\bar{z})(x+\bar{y}+z)(x+\bar{y}+\bar{z})$ **d)** $(x+y+z)(x+y+\bar{z})(x+\bar{y}+z)(x+\bar{y}+\bar{z})(\bar{x}+\bar{y}+z)(\bar{x}+\bar{y}+\bar{z})$

13. **a)** $x + y + z$ **b)** $x + \overline{[y + (\bar{x} + z)]}$ **c)** $\overline{(x + \bar{y})}$ **d)** $\overline{[x + \overline{(x + \bar{y} + \bar{z})}]}$

15. **a)**

x	$\bar{x}$	$x \downarrow x$
1	0	0
0	1	1

b)

x	y	xy	$x \downarrow x$	$y \downarrow y$	$(x \downarrow x) \downarrow (y \downarrow y)$
1	1	1	0	0	1
1	0	0	0	1	0
0	1	0	1	0	0
0	0	0	1	1	0

c)

x	y	$x+y$	$(x \downarrow y)$	$(x \downarrow y) \downarrow (x \downarrow y)$
1	1	1	0	1
1	0	1	0	1
0	1	1	0	1
0	0	0	1	0

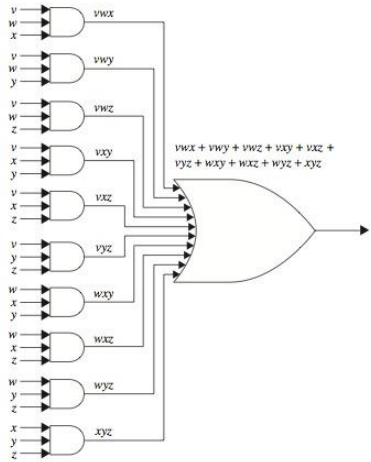
17. **a)** $\{[(x \mid x) \mid (y \mid y)] \mid [(x \mid x) \mid (y \mid y)]\} \mid (z \mid z)$ **b)** $\{[(x \mid x) \mid (z \mid z)] \mid y\} \mid \{[(x \mid x) \mid (z \mid z)] \mid y\}$ **c)** x **d)** $[x \mid (y \mid y)] \mid [x \mid (y \mid y)]$ **19.** É impossível representar

$\bar{x}$ usando + e · porque não existe nenhuma maneira de obter o valor 0 se a entrada for 1.

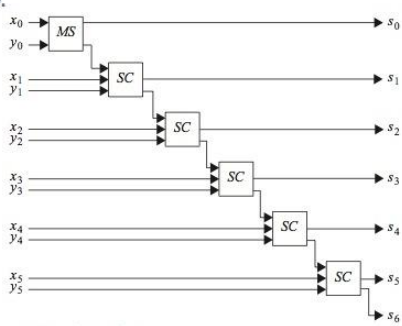
Seção 11.3

1. $(x + y)\bar{y}$ **3.** $\overline{(xy)} + (\bar{z} + x)$ **5.** $(x + y + z) + (\bar{x} + y + z) + (\bar{x} + \bar{y} + \bar{z})$

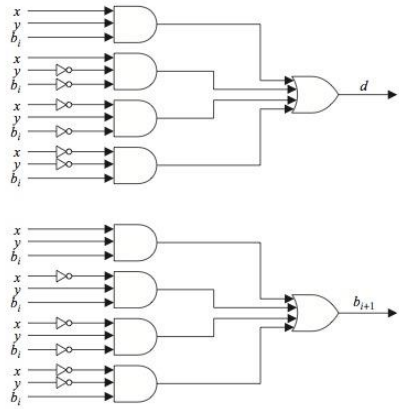
7.



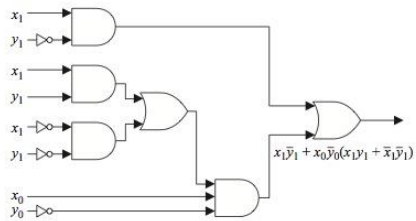
9.



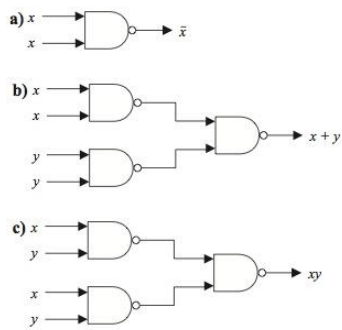
11.



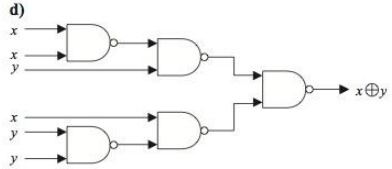
13.



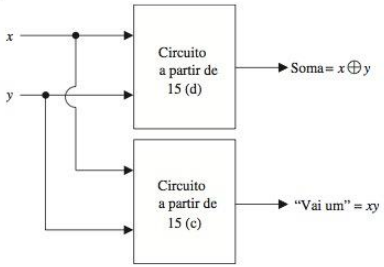
15.



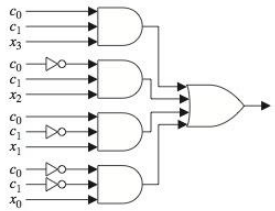
d)



17.

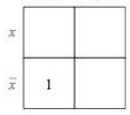


19.



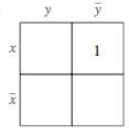
Seção 11.4

1. a)

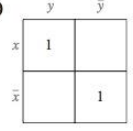


b) xy e $x\bar{y}$

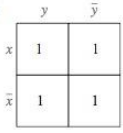
3. a)



b)



c)



5. a)

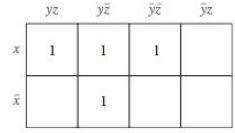
b) $\bar{x}yz$, $\bar{x}y\bar{z}$, $xy\bar{z}$

7. a)

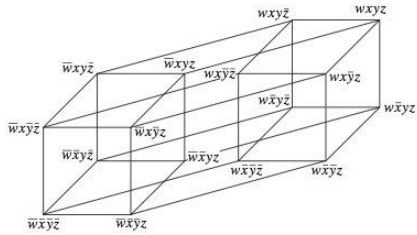
b)

c)

9. Implicantes: xyz, $xy\bar{z}$, $x\bar{y}\bar{z}$, $\bar{x}y\bar{z}$, xy, $x\bar{z}$, $y\bar{z}$; implicantes primos: xy, $x\bar{z}$, $y\bar{z}$; implicantes primos essenciais: xy, $x\bar{z}$, $y\bar{z}$



11. O 3-cubo na direita corresponde a w; o 3-cubo dado pela superfície de cima da figura toda representa x; o 3-cubo dado pela parte de trás da figura toda representa y; o 3-cubo dado pelas superfícies direitas de ambos os 3-cubos direito e esquerdo representa z. Em cada caso, a 3-face oposta representa o literal complementado. O 2-cubo que representa wz é a face direita do 3-cubo à direita; o 2-cubo que representa $\bar{x}y$ é a parte de trás de baixo; o 2-cubo que representa $\bar{y}\bar{z}$ é a parte esquerda da frente.



13. a)

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx				
$w\bar{x}$				
$\bar{w}\bar{x}$				
$\bar{w}x$		1		

b) $\bar{w}xyz$, $\bar{w}x\bar{y}z$, $\bar{w}xy\bar{z}$, $\bar{w}x\bar{y}\bar{z}$

15. a)

	$x_3x_4x_5$	$x_3x_4\bar{x}_5$	$x_3\bar{x}_4\bar{x}_5$	$x_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4\bar{x}_5$	$\bar{x}_3x_4\bar{x}_5$	$\bar{x}_3x_4x_5$
x_1x_2	1	1						
$x_1\bar{x}_2$								
$\bar{x}_1\bar{x}_2$								
$\bar{x}_1x_2$								

b)

	$x_3x_4x_5$	$x_3x_4\bar{x}_5$	$x_3\bar{x}_4\bar{x}_5$	$x_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4\bar{x}_5$	$\bar{x}_3x_4\bar{x}_5$	$\bar{x}_3x_4x_5$
x_1x_2								
$x_1\bar{x}_2$								
$\bar{x}_1\bar{x}_2$	1			1				
$\bar{x}_1x_2$	1			1				

c)

	$x_3x_4x_5$	$x_3x_4\bar{x}_5$	$x_3\bar{x}_4\bar{x}_5$	$x_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4\bar{x}_5$	$\bar{x}_3x_4\bar{x}_5$	$\bar{x}_3x_4x_5$
x_1x_2	1	1					1	1
$x_1\bar{x}_2$								
$\bar{x}_1\bar{x}_2$								
$\bar{x}_1x_2$	1	1					1	1

d)

	$x_3x_4x_5$	$x_3x_4\bar{x}_5$	$x_3\bar{x}_4\bar{x}_5$	$x_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4\bar{x}_5$	$\bar{x}_3x_4\bar{x}_5$	$\bar{x}_3x_4x_5$
x_1x_2					1	1		
$x_1\bar{x}_2$					1	1		
$\bar{x}_1\bar{x}_2$					1	1		
$\bar{x}_1x_2$					1	1		

e)

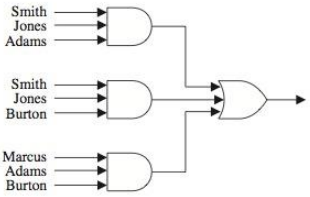
	$x_3x_4x_5$	$x_3x_4\bar{x}_5$	$x_3\bar{x}_4\bar{x}_5$	$x_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4\bar{x}_5$	$\bar{x}_3x_4\bar{x}_5$	$\bar{x}_3x_4x_5$
x_1x_2	1	1	1	1				
$x_1\bar{x}_2$	1	1	1	1				
$\bar{x}_1\bar{x}_2$	1	1	1	1				
$\bar{x}_1x_2$	1	1	1	1				

f)

	$x_3x_4x_5$	$x_3x_4\bar{x}_5$	$x_3\bar{x}_4\bar{x}_5$	$x_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4x_5$	$\bar{x}_3\bar{x}_4\bar{x}_5$	$\bar{x}_3x_4\bar{x}_5$	$\bar{x}_3x_4x_5$
x_1x_2		1	1			1	1	
$x_1\bar{x}_2$		1	1			1	1	
$\bar{x}_1\bar{x}_2$		1	1			1	1	
$\bar{x}_1x_2$		1	1			1	1	

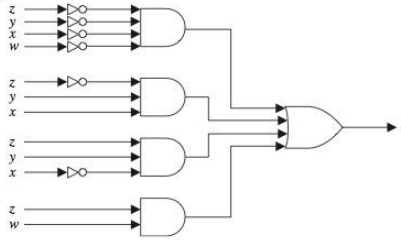
17. **a)** 64 **b)** 6 **19.** As linhas 1 e 4 são consideradas adjacentes. Os pares de colunas considerados adjacentes são: colunas 1 e 4, 1 e 12, 1 e 16, 2 e 11, 2 e 15, 3 e 6, 3 e 10, 4 e 9, 5 e 8, 5 e 16, 6 e 15, 7 e 10, 7 e 14, 8 e 13, 9 e 12, 11 e 14, 13 e 16.

21.



23. **a)** $\bar{x}z$ **b)** y **c)** $x\bar{z} + \bar{x}z + \bar{y}z$ **d)** $xz + \bar{x}y + \bar{y}\bar{z}$
25. **a)** $wxz + wx\bar{y} + w\bar{y}z + w\bar{x}\bar{y}\bar{z}$ **b)** $x\bar{y}\bar{z} + \bar{w}\bar{y}z + wxy\bar{z} + w\bar{x}yz + \bar{w}\bar{x}\bar{y}\bar{z}$ **c)** $\bar{y}z + wxy + w\bar{x}\bar{y} + \bar{w}\bar{x}\bar{y}\bar{z}$
d) $wy + yz + \bar{x}y + wxz + \bar{w}\bar{x}z$ **27.** $x(y + z)$

29.



31. $x\bar{z} + xz$ **33.** Usamos indução em n. Se $n = 1$, então estamos olhando para um segmento de reta, rotulado 0 em uma extremidade e 1 na outra extremidade. O único valor possível de k também é 1, e, se o literal for x_1, então o subcubo que temos é o subcubo 0-dimensional que consiste nas extremidades rotuladas 1, e, se o literal for $\bar{x}_1$, então o subcubo que temos é o subcubo 0-dimensional que consiste nas extremidades rotuladas 0. Suponha agora que a afirmação seja verdadeira para n; devemos mostrar que ela é verdadeira para $n + 1$. Se o literal x_{n+1} (ou seu complemento) não for uma parte do produto, então, pela hipótese de indução, o produto, quando visto no contexto de n variáveis, corresponde a um subcubo $(n - k)$-dimensional do cubo n-dimensional, e o produto cartesiano daquele subcubo com o segmento de reta [0, 1] nos dá um subcubo com uma dimensão a mais no cubo $(n + 1)$-dimensional dado, ou seja, que tem dimensão $(n + 1) - k$, como desejado. Por outro lado, se o literal x_{n+1} (ao seu complemento) for parte do produto, então o produto dos $k - 1$ literais restantes corresponde a um subcubo de dimensão $n - (k - 1) = (n + 1) - k$ no cubo n-dimensional, e aquela fatia, ou na extremidade 1 ou na extremidade 0 na última variável, é o subcubo desejado.

Exercícios Complementares

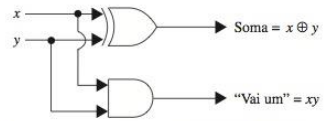
1. **a)** $x = 0, y = 0, z = 0$; $x = 1, y = 1, z = 1$ **b)** $x = 0, y = 0, z = 0$; $x = 0, y = 0, z = 1$; $x = 0, y = 1, z = 0$; $x = 1, y = 0, z = 1$; $x = 1, y = 1, z = 0$; $x = 1, y = 1, z = 1$ **c)** Nenhum valor. **3.** **a)** Sim. **b)** Não. **c)** Não. **d)** Sim. **5.** $2^{2^{n-1}}$ **7.** **a)** Se $F(x_1, ..., x_n) = 1$, então $(F + G)(x_1, ..., x_n) = F(x_1, ..., x_n) + G(x_1, ..., x_n) = 1$ pela propriedade de dominação. Logo, $F \leq F + G$. **b)** Se $(FG)(x_1, ..., x_n) = 1$, então $F(x_1, ..., x_n) \cdot G(x_1, ..., x_n) = 1$. Logo, $F(x_1, ..., x_n) = 1$. Segue que $FG \leq F$. **9.** Como $F(x_1, ..., x_n) = 1$ implica que $F(x_1, ..., x_n) = 1$, $\leq$ é reflexiva. Suponha que $F \leq G$ e $G \leq F$. Então $F(x_1, ..., x_n) = 1$ se e somente se $G(x_1, ..., x_n) = 1$. Isto implica que $F = G$. Portanto, $\leq$ é anti-simétrica. Suponha que $F \leq G \leq H$. Então, se $F(x_1, ..., x_n) = 1$, segue que $G(x_1, ..., x_n) = 1$, o que implica que $H(x_1, ..., x_n) = 1$. Logo, $F \leq H$, de modo que $\leq$ é transitiva. **11.** **a)** $x = 1, y = 0, z = 0$ **b)** $x = 1, y = 0, z = 0$ **c)** $x = 1, y = 0, z = 0$

13.

x	y	$x \odot y$	$x \oplus y$	$\overline{(x \oplus y)}$
1	1	1	0	1
1	0	0	1	0
0	1	0	1	0
0	0	1	0	1

15. Sim, como mostra uma tabela-verdade. **17.** **a)** 6 **b)** 5 **c)** 5 **d)** 6

19.



21. $x_3 + x_2\bar{x}_1$ **23.** Suponha que ele tivesse os pesos a e b. Então, existiria um número real T, tal que $xa + yb \geq T$ para (1, 0) e (0, 1), mas com $xa + yb < T$ para (0, 0) e (1, 1). Logo, $a \geq T$, $b \geq T$, $0 < T$ e $a + b < T$. Assim, a e b são positivos, o que implica que $a + b > a \geq T$, uma contradição.

CAPÍTULO 12

Seção 12.1

1. **a)** **sentença ⇒ predicado nominal predicado verbal intransitivo ⇒ artigo adjetivo substantivo predicado verbal intransitivo ⇒ artigo adjetivo substantivo verbo intransitivo** ⇒ ... (depois de 3 passos) ... ⇒ *a alegre lebre corre.*
b) **sentença ⇒ predicado nominal predicado verbal intransitivo ⇒ artigo adjetivo substantivo predicado verbal intransitivo ⇒ artigo adjetivo substantivo verbo intransitivo advérbio** ... (depois de 4 passos) ... ⇒ *a sonolenta tartaruga corre rapidamente*

c) sentença ⇒ predicado nominal predicado verbal transitivo predicado nominal ⇒ artigo substantivo predicado verbal transitivo predicado nominal ⇒ artigo substantivo verbo transitivo predicado nominal ⇒ artigo substantivo verbo transitivo artigo substantivo ⇒ ...(depois de 4 passos) ... ⇒ *a tartaruga passa a lebre*

d) sentença ⇒ predicado nominal predicado verbal transitivo predicado nominal ⇒ artigo adjetivo substantivo predicado verbal transitivo predicado nominal ⇒ artigo adjetivo substantivo verbo transitivo predicado nominal ⇒ artigo adjetivo substantivo verbo transitivo artigo adjetivo substantivo ⇒ ...(depois de 6 passos) ... ⇒ *a sonolenta lebre passa a alegre tartaruga*

3. A única maneira de obter um substantivo, como *tartaruga*, no final é ter um predicado nominal no final, o que pode ser alcançado apenas pela produção **sentença → predicado nominal predicado verbal transitivo predicado nominal**. Entretanto, **predicado verbal transitivo → verbo transitivo** → *passa*, e esta sentença não contém *passa*.

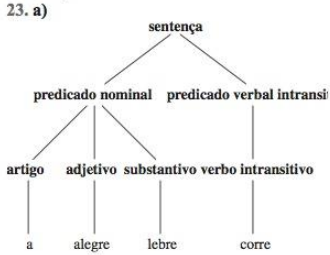
5. a) $S \Rightarrow 1A \Rightarrow 10B \Rightarrow 101A \Rightarrow 1010B \Rightarrow 10101$ **b)** Por causa das regras de produção desta gramática, todo 1 deve ser seguido por um 0, a menos que ele ocorra no final da seqüência. **c)** Todas as seqüências que consistem em um 0 ou um 1 seguidos por uma ou mais repetições de 01.

7. $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000S111 \Rightarrow 000111$

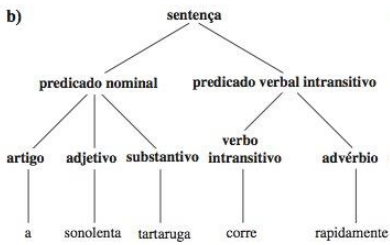
9. a) $S \Rightarrow 0S \Rightarrow 00S \Rightarrow 00S1 \Rightarrow 00S11 \Rightarrow 00S111 \Rightarrow 00S1111 \Rightarrow 001111$ **b)** $S \Rightarrow 0S \Rightarrow 00S \Rightarrow 001A \Rightarrow 0011A \Rightarrow 00111A \Rightarrow 001111$ **11.** $S \Rightarrow 0SAB \Rightarrow 00SABAB \Rightarrow 00ABAB \Rightarrow 00AABB \Rightarrow 001ABB \Rightarrow 0011BB \Rightarrow 00112B \Rightarrow 001122$

13. a) $S \to 0, S \to 1, S \to 11$ **b)** $S \to 1S, S \to \lambda$ **c)** $S \to 0A1, A \to 1A, A \to 0A, A \to \lambda$ **d)** $S \to 0A, A \to 11A, A \to \lambda$. **15. a)** $S \to 00S, S \to \lambda$ **b)** $S \to 10A, A \to 00A, A \to \lambda$ **c)** $S \to AAS, S \to BBS, AB \to BA, BA \to AB, S \to \lambda, A \to 0, B \to 1$ **d)** $S \to 0000000000A, A \to 0A, A \to \lambda$ **e)** $S \to AS, S \to ABS, S \to A, AB \to BA, BA \to AB, A \to 0, B \to 1$ **f)** $S \to ABS, S \to \lambda, AB \to BA, BA \to AB, A \to 0, B \to 1$. **g)** $S \to ABS, S \to T, S \to U, T \to AT, T \to A, U \to BU, U \to B, AB \to BA, BA \to AB, A \to 0, B \to 1$ **17. a)** $S \to 0S, S \to \lambda$ **b)** $S \to A0, A \to 1A, A \to \lambda$ **c)** $S \to 000S, S \to \lambda$ **19. a)** Tipo 2, não é tipo 3. **b)** Tipo 3. **c)** Tipo 0, não é tipo 1. **d)** Tipo 2, não é tipo 3. **e)** Tipo 2, não é tipo 3. **f)** Tipo 0, não é tipo 1. **g)** Tipo 3. **h)** Tipo 0, não é tipo 1. **i)** Tipo 2, não é tipo 3. **j)** Tipo 2, não é tipo 3. **21.** Sejam S_1 e S_2 os símbolos iniciais de G_1 e G_2, respectivamente. Seja S um novo símbolo inicial. **a)** Adicione S e as regras de produção $S \to S_1$ e $S \to S_2$. **b)** Adicione S e as regras de produção $S \to S_1 S_2$. **c)** Adicione S e as regras de produção $S \to \lambda$ e $S \to S_1 S$.

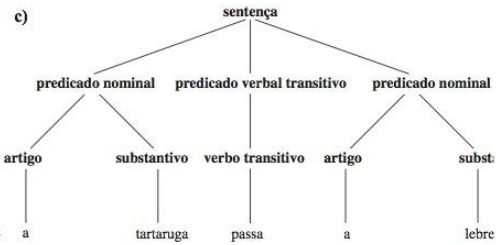
23. a)



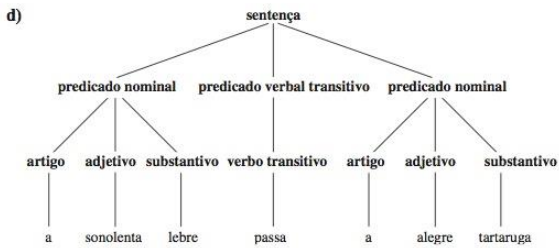
b)



c)

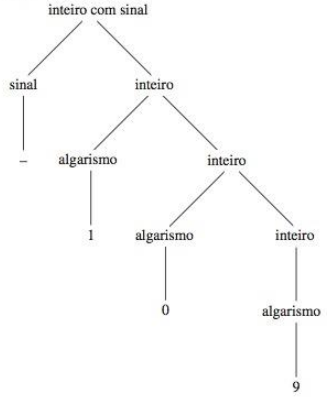


d)



25. **a)** Sim. **b)** Não. **c)** Sim. **d)** Não.

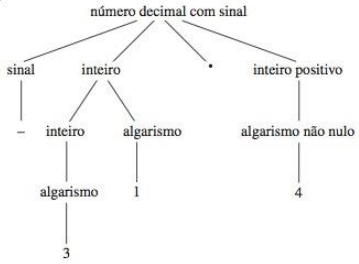
27.



29. **a)** $S \rightarrow \langle sinal \rangle \langle inteiro \rangle$
$S \rightarrow \langle sinal \rangle \langle inteiro \rangle . \langle inteiro\ positivo \rangle$
$\langle sinal \rangle \rightarrow +$
$\langle sinal \rangle \rightarrow -$
$\langle inteiro \rangle \rightarrow \langle algarismo \rangle$
$\langle inteiro \rangle \rightarrow \langle inteiro \rangle \langle algarismo \rangle$
$\langle algarismo \rangle \rightarrow i, i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 0$
$\langle inteiro\ positivo \rangle \rightarrow \langle inteiro \rangle \langle algarismo\ não\ nulo \rangle$
$\langle inteiro\ positivo \rangle \rightarrow \langle algarismo\ não\ nulo \rangle \langle inteiro \rangle$
$\langle inteiro\ positivo \rangle \rightarrow \langle inteiro \rangle \langle algarismo\ não\ nulo \rangle \langle inteiro \rangle$
$\langle inteiro\ positivo \rangle \rightarrow \langle algarismo\ não\ nulo \rangle$
$\langle algarismo\ não\ nulo \rangle \rightarrow i, i = 1, 2, 3, 4, 5, 6, 7, 8, 9$

b) $\langle número\ decimal\ com\ sinal \rangle ::= \langle sinal \rangle \langle inteiro \rangle \mid \langle sinal \rangle \langle inteiro \rangle . \langle inteiro\ positivo \rangle$
$\langle sinal \rangle ::= + \mid -$
$\langle inteiro \rangle ::= \langle algarismo \rangle \mid \langle inteiro \rangle \langle algarismo \rangle$
$\langle algarismo \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
$\langle algarismo\ não\ nulo \rangle ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
$\langle inteiro\ positivo \rangle ::= \langle inteiro \rangle \langle algarismo\ não\ nulo \rangle \mid \langle algarismo\ não\ nulo \rangle \langle inteiro \rangle \mid \langle inteiro \rangle$
$\langle inteiro\ não\ nulo \rangle \langle inteiro \rangle \mid \langle algarismo\ não\ nulo \rangle$

c)



31. **a)** $\langle identificador \rangle ::= \langle letramin \rangle \mid \langle identificador \rangle \langle letramin \rangle$
$\langle letramin \rangle ::= a \mid b \mid c \mid \cdots \mid z$

b) $\langle identificador \rangle ::= \langle letramin \rangle \langle letramin \rangle \langle letramin \rangle \mid \langle letramin \rangle \langle letramin \rangle \langle letramin \rangle \langle letramin \rangle \mid \langle letramin \rangle \langle letramin \rangle \langle letramin \rangle \langle letramin \rangle \langle letramin \rangle \mid \langle letramin \rangle \langle letramin \rangle \langle letramin \rangle \langle letramin \rangle \langle letramin \rangle \langle letramin \rangle$
$\langle letramin \rangle ::= a \mid b \mid c \mid \cdots \mid z$

c) $\langle identificador \rangle ::= \langle letramaiu \rangle \mid \langle letramaiu \rangle \langle letra \rangle \mid \langle letramaiu \rangle \langle letra \rangle \langle letra \rangle \mid \langle letramaiu \rangle \langle letra \rangle \langle letra \rangle \langle letra \rangle \mid \langle letramaiu \rangle \langle letra \rangle \langle letra \rangle \langle letra \rangle \mid \langle letramaiu \rangle \langle letra \rangle \langle letra \rangle \langle letra \rangle \langle letra \rangle \langle letra \rangle$
$\langle letra \rangle ::= \langle letramin \rangle \mid \langle letramaiu \rangle$
$\langle letramin \rangle ::= a \mid b \mid c \mid \cdots \mid z$
$\langle letramaiu \rangle ::= A \mid B \mid C \mid \cdots \mid Z$

d) $\langle identificador \rangle ::= \langle letramin \rangle \langle algarismoouunder \rangle \langle alfanumérico \rangle \langle alfanumérico \rangle \langle alfanumérico \rangle \mid \langle letramin \rangle \langle algarismoouunder \rangle \langle alfanumérico \rangle \langle alfanumérico \rangle \langle alfanumérico \rangle \langle alfanumérico \rangle$
$\langle algarismoouunder \rangle ::= \langle algarismo \rangle \mid -$
$\langle alfanumérico \rangle ::= \langle letra \rangle \mid \langle algarismo \rangle$
$\langle letra \rangle ::= \langle letramin \rangle \mid \langle letramaiu \rangle$
$\langle letramin \rangle ::= a \mid b \mid c \mid \cdots \mid z$
$\langle letramaiu \rangle ::= A \mid B \mid C \mid \cdots \mid Z$
$\langle algarismo \rangle ::= 0 \mid 1 \mid 2 \mid \cdots \mid 9$

33. $\langle identificador \rangle ::= \langle letraouus \rangle \mid \langle identificador \rangle \langle símbolo \rangle$
$\langle letraouus \rangle ::= \langle letra \rangle \mid -$
$\langle símbolo \rangle ::= \langle letraouus \rangle \mid \langle algarismo \rangle$
$\langle letra \rangle ::= \langle letramin \rangle \mid \langle letramaiu \rangle$
$\langle letramin \rangle ::= a \mid b \mid c \mid \cdots \mid z$
$\langle letramaiu \rangle ::= A \mid B \mid C \mid \cdots \mid Z$
$\langle algarismo \rangle ::= 0 \mid 1 \mid 2 \mid \cdots \mid 9$

35. *numeral ::= sinal? algarismonãonulo algarismo* decimal? | sinal? 0 decimal?*
sinal ::= + | −
algarismonãonulo ::= 1 | 2 | ··· | 9
algarismo ::= 0 | algarismonãonulo
*decimal ::=.algarismo**

37. *identificador ::= letraous símbolo**
letraounder ::= letra | -
símbolo ::= letraouus | algarismo
letra ::= letramin | letramaiu
letramin ::= a | b | c | ··· | z
letramaiu ::= A | B | C | ··· | Z
algarismo ::= 0 | 1 | 2 | ··· | 9

39. **a)** $\langle expressão \rangle$
$\langle termo \rangle \langle termo \rangle \langle Operador\ de\ soma \rangle$
$\langle fator \rangle \langle fator \rangle \langle fator \rangle \langle Operador\ de\ multiplicação \rangle \langle Operador\ de\ soma \rangle$
$\langle identificador \rangle \langle identificador \rangle \langle identificador \rangle \langle Operador\ de\ multiplicação \rangle \langle Operador\ de\ soma \rangle$
*a b c * +*

b) Não é gerada.
c) ⟨*expressão*⟩
⟨*termo*⟩
⟨*fator*⟩ ⟨*fator*⟩ ⟨*Operador de multiplicação*⟩
⟨*expressão*⟩ ⟨*fator*⟩ ⟨*Operador de multiplicação*⟩
⟨*termo*⟩ ⟨*termo*⟩ ⟨*Operador de soma*⟩ ⟨*fator*⟩ ⟨*Operador de multiplicação*⟩
⟨*fator*⟩ ⟨*fator*⟩ ⟨*Operador de soma*⟩ ⟨*fator*⟩ ⟨*Operador de multiplicação*⟩
⟨*identificador*⟩ ⟨*identificador*⟩ ⟨*Operador de soma*⟩ ⟨*identificador*⟩ ⟨*Operador de multiplicação*⟩
$xy - z *$
d) ⟨*expressão*⟩
⟨*termo*⟩
⟨*fator*⟩ ⟨*fator*⟩ ⟨*Operador de multiplicação*⟩
⟨*fator*⟩ ⟨*expressão*⟩ ⟨*Operador de multiplicação*⟩
⟨*fator*⟩ ⟨*termo*⟩ ⟨*Operador de multiplicação*⟩
⟨*fator*⟩ ⟨*fator*⟩ ⟨*fator*⟩ ⟨*Operador de multiplicação*⟩ ⟨*Operador de multiplicação*⟩
⟨*fator*⟩ ⟨*fator*⟩ ⟨*expressão*⟩ ⟨*Operador de multiplicação*⟩ ⟨*Operador de multiplicação*⟩
⟨*fator*⟩ ⟨*fator*⟩ ⟨*termo*⟩ ⟨*termo*⟩ ⟨*Operador de soma*⟩ ⟨*Operador de multiplicação*⟩ ⟨*Operador de multiplicação*⟩
⟨*fator*⟩ ⟨*fator*⟩ ⟨*fator*⟩ ⟨*fator*⟩ ⟨*Operador de soma*⟩ ⟨*Operador de multiplicação*⟩ ⟨*Operador de multiplicação*⟩
⟨*identificador*⟩ ⟨*identificador*⟩ ⟨*identificador*⟩ ⟨*identificador*⟩ ⟨*Operador de soma*⟩ ⟨*Operador de multiplicação*⟩
⟨*Operador de multiplicação*⟩
$wxyz - * /$
e) ⟨*expressão*⟩
⟨*termo*⟩
⟨*fator*⟩ ⟨*fator*⟩ ⟨*Operador de multiplicação*⟩
⟨*fator*⟩ ⟨*expressão*⟩ ⟨*Operador de multiplicação*⟩
⟨*fator*⟩ ⟨*termo*⟩ ⟨*termo*⟩ ⟨*Operador de soma*⟩ ⟨*Operador de multiplicação*⟩
⟨*fator*⟩ ⟨*fator*⟩ ⟨*fator*⟩ ⟨*Operador de soma*⟩ ⟨*Operador de multiplicação*⟩
⟨*identificador*⟩ ⟨*identificador*⟩ ⟨*identificador*⟩ ⟨*Operador de soma*⟩ ⟨*Operador de multiplicação*⟩
$ade - *$

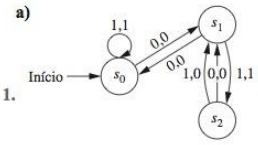
41. a) Não é gerada.
b) ⟨*expressão*⟩
⟨*termo*⟩ ⟨*Operador de soma*⟩ ⟨*termo*⟩
⟨*fator*⟩ ⟨*Operador de multiplicação*⟩ ⟨*fator*⟩ ⟨*Operador de soma*⟩ ⟨*fator*⟩ ⟨*Operador de multiplicação*⟩ ⟨*fator*⟩
⟨*identificador*⟩ ⟨*Operador de multiplicação*⟩ ⟨*identificador*⟩ ⟨*Operador de soma*⟩ ⟨*identificador*⟩ ⟨*Operador de multiplicação*⟩ ⟨*identificador*⟩
$a/b + c/d$
c) ⟨*expressão*⟩
⟨*termo*⟩
⟨*fator*⟩ ⟨*Operador de multiplicação*⟩ ⟨*fator*⟩
⟨*fator*⟩ ⟨*Operador de multiplicação*⟩ (⟨*expressão*⟩)
⟨*fator*⟩ ⟨*Operador de multiplicação*⟩ (⟨*termo*⟩ ⟨*Operador de soma*⟩ ⟨*termo*⟩)
⟨*fator*⟩ ⟨*Operador de multiplicação*⟩ (⟨*fator*⟩ ⟨*Operador de soma*⟩ ⟨*fator*⟩)
⟨*identificador*⟩ ⟨*Operador de multiplicação*⟩ (⟨*identificador*⟩ ⟨*Operador de soma*⟩ ⟨*identificador*⟩)
$m * (n + p)$
d) Não é gerada.
e) ⟨*expressão*⟩
⟨*termo*⟩
⟨*fator*⟩ ⟨*Operador de multiplicação*⟩ ⟨*fator*⟩
(⟨*expressão*⟩) ⟨*Operador de multiplicação*⟩ (⟨*expressão*⟩)
(⟨*termo*⟩ ⟨*Operador de soma*⟩⟨*termo*⟩) ⟨*Operador de multiplicação*⟩ (⟨*termo*⟩ ⟨*Operador de soma*⟩ ⟨*termo*⟩)

⟨*fator*⟩ ⟨*Operador de soma*⟩ ⟨*fator*⟩) ⟨*Operador de multiplicação*⟩ (⟨*fator*⟩ ⟨*Operador de soma*⟩⟨*fator*⟩)
(⟨*identificador*⟩ ⟨*Operador de soma*⟩ ⟨*identificador*⟩)
⟨*Operador de multiplicação*⟩ (⟨*identificador*⟩ ⟨*Operador de soma*⟩ ⟨*identificador*⟩)
$(m + n) * (p - q)$

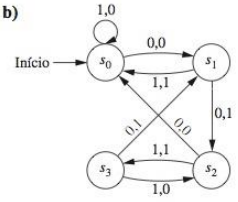
Seção 12.2

1.

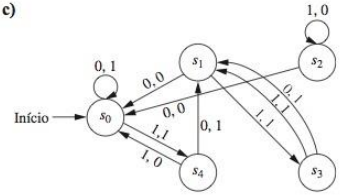
a)



b)

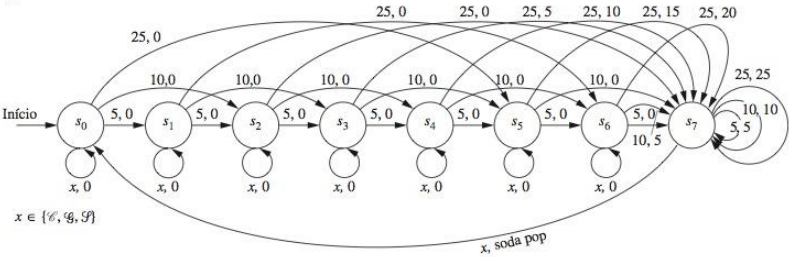


c)



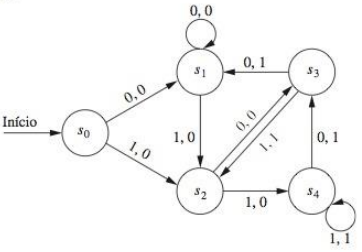
3. **a)** 01010 **b)** 01000 **c)** 11011 **5.** **a)** 1100 **b)** 00110110 **c)** 11111111111

7.

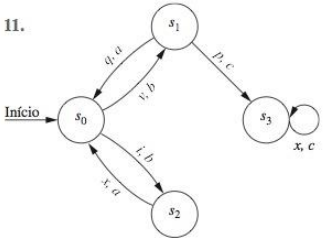


$\mathscr{C}$ = coca-cola
$\mathscr{G}$ = guaraná
$\mathscr{S}$ = soda-limonada

9.

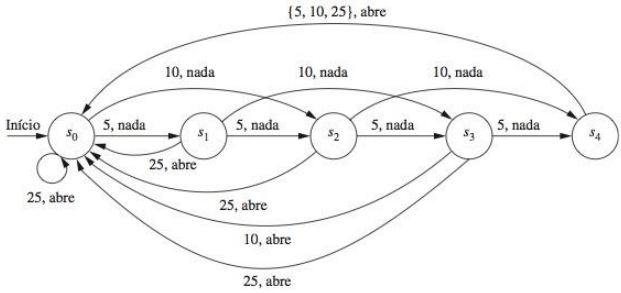


11.



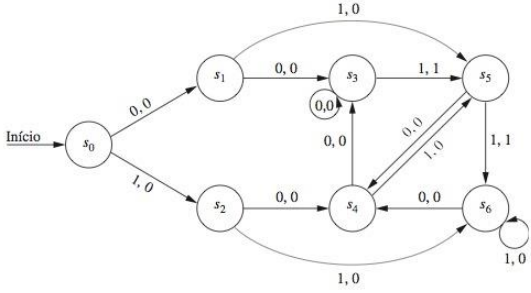
v = ID válido
i = ID inválido
p = Senha válida
q = Senha inválida
a = "Entre o ID do usuário"
b = "Entre senha"
c = Prompt
x = Qualquer entrada

13.

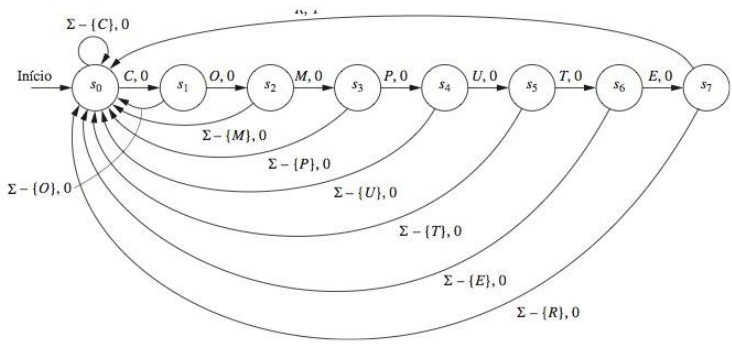


15. Seja s_0 o estado inicial e seja s_1 o estado que representa uma chamada bem-sucedida. A partir de s_0, entradas 2, 3, 4, 5, 6, 7 ou 8 mandam a máquina de volta para s_0 com uma mensagem de erro para o usuário como saída. A partir de s_0, uma entrada 0 manda a máquina para o estado s_1, com a saída, sendo que o 0 é mandado para a rede. A partir de s_0, uma entrada 9 manda a máquina para o estado s_2 sem nenhuma saída; a partir de lá, uma entrada 1 manda a máquina para o estado s_3 sem nenhuma saída; a partir de lá, uma entrada 1 manda a máquina para o estado s_1 com a saída, sendo que 911 é mandado para a rede. Todas as outras entradas, enquanto nos estados s_2 ou s_3, mandam a máquina de volta para s_0 com uma mensagem de erro para o usuário como saída. A partir de s_0, uma entrada 1 manda a máquina para o estado s_4 sem nenhuma saída; a partir de s_4, uma entrada 2 manda a máquina para o estado s_5 sem nenhuma saída; e este caminho continua de modo parecido com o caminho do 911, procurando o próximo 1, depois 2, depois quaisquer sete algarismos, em cujo ponto a máquina vai para o estado s_1 com a saída, sendo que a entrada de dez algarismos é mandada para a rede. Qualquer entrada "incorreta" enquanto nos estados s_5 ou s_6 (isto é, qualquer coisa, exceto um 1 enquanto em s_5 ou um 2 enquanto em s_6) manda a máquina de volta para s_0 com uma mensagem de erro para o usuário como saída. Analogamente, a partir de s_4 uma entrada 8 seguida por sucessores adequados leva eventualmente a s_1, mas saídas inadequadas nos levam de volta para s_0 com uma mensagem de erro. Além disso, saídas, enquanto no estado s_4, diferentes de 2 ou 8 mandam a máquina de volta para o estado s_0 com uma mensagem de erro para o usuário como saída.

17.



19.

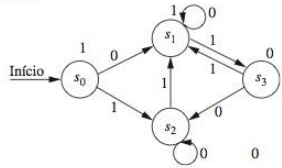


21.

	f		
	Entrada		
Estado	**0**	**1**	***g***
s_0	s_1	s_2	1
s_1	s_1	s_0	1
s_2	s_1	s_2	0

23. **a)** 11111
b) 1000000
c) 100011001100

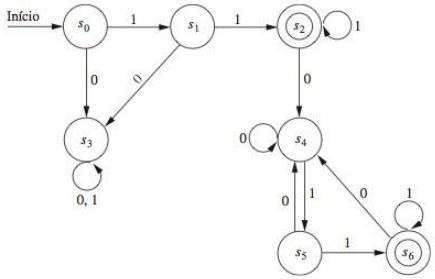
25.



Seção 12.3

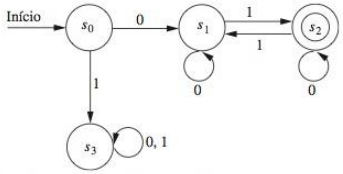
1. **a)** {000, 001, 1100, 1101} **b)** {000, 0011, 010, 0111} **c)** {00, 011, 110, 1111} **d)** {000000, 000001, 000100, 000101, 010000, 010001, 010100, 010101} **3.** $A = \{1, 101\}$, $B = \{0, 11, 000\}$; $A = \{10, 111, 1010, 1000, 10111, 101000\}$, $B = \{\lambda\}$; $A = \{\lambda, 10\}$, $B = \{10, 111, 1000\}$ ou $A = \{\lambda\}$, $B = \{10, 111, 1010, 1000, 10111, 101000\}$ **5.** **a)** O conjunto de seqüências que consiste em zero ou mais pares de bits consecutivos 10. **b)** O conjunto de seqüências que consiste apenas em 1s, tal que o número de 1s seja divisível por 3, incluindo a seqüência nula. **c)** O conjunto de seqüências nas quais todo 1 é imediatamente precedido por um 0. **d)** O conjunto de seqüências que começam e terminam com um 1 e têm pelo menos dois 1s entre todo par de 0s. **7.** Uma seqüência está em A^* se e somente se for uma concatenação de um número arbitrário de seqüências em A. Como cada seqüência em A também está em B, segue que uma seqüência em A^* também é uma concatenação de seqüências em B. Logo, $A^* \subseteq B^*$. **9.** **a)** Sim. **b)** Sim. **c)** Não. **d)** Não. **e)** Sim. **f)** Sim. **11.** **a)** Sim. **b)** Não. **c)** Sim. **d)** Não. **13.** **a)** Sim. **b)** Sim. **c)** Não. **d)** Não. **e)** Não. **f)** Não **15.** Usamos indução estrutural na seqüência de entrada y. O passo base considera $y = \lambda$, e para o passo de indução escrevemos $y = wa$, em que $w \in I^*$ e $a \in I$. Para o passo base, temos $xy = x$, de modo que devemos mostrar que $f(s, x) = f(f(s, x), \lambda)$. Mas a parte (i) da definição de função de transição estendida diz que isto é verdadeiro. Supomos então a hipótese de indução, em que a equação é válida para w e demonstramos que $f(s, xwa) = f(f(s, x), wa)$. Pela parte (ii) da definição, o lado esquerdo desta equação é igual a $f(f(s, xw), a)$. Pela hipótese de indução, $f(s, xw) = f(f(s, x), w)$, de modo que $f(f(s, xw), a) = f(f(f(s, x), w), a)$. O lado direito da igualdade que queremos, pela parte (ii) da definição, também é igual a $f(f(f(s, x), w), a)$, como desejado. **17.** $\{0, 10, 11\}\{0, 1\}^*$ **19.** $\{0^m 1^n \mid m \geq 0 \text{ e } n \geq 1\}$ **21.** $\{\lambda\} \cup \{0\}\{1\}^*\{0\} \cup \{10, 11\}\{0, 1\}^* \cup \{0\}\{1\}^*\{01\}\{0, 1\}^* \cup \{0\}\{1\}^*\{00\}\{0\}^*\{1\}\{0, 1\}^*$ **23.** Seja s_2 o único estado final, e ponha transições a partir de s_2 para si próprio em uma das entradas. Ponha uma transição a partir do estado inicial s_0 para s_1 com entrada 0, e uma transição a partir de s_1 para s_2 com entrada 1. Crie o estado s_3 e faça com que as outras transições a partir de s_0 e s_1 (bem como ambas as transições a partir de s_3) levem a s_3. **25.** Estado inicial s_0, único estado final s_3; transições a partir de s_0 para s_0 com entrada 0, a partir de s_0 para s_1 com entrada 1, a partir de s_1 para s_2 com entrada 0, a partir de s_1 para s_1 com entrada 1, a partir de s_2 para s_0 com entrada 0, a partir de s_2 para s_3 com entrada 1, a partir de s_3 para s_3 com entrada 0, a partir de s_3 para s_3 com entrada 1. **27.** Considere cinco estados, com apenas s_3 sendo final. Para $i = 0, 1, 2, 3$, transição a partir de s_i para si mesmo com entrada 1 e para s_{i+1} com entrada 0. Ambas as transições a partir de s_4 são para ele mesmo. **29.** Considere quatro estados, com apenas s_3 sendo final. Para $i = 0, 1, 2$, transição a partir de s_i para s_{i+1} com entrada 1 mas de volta para s_0 com entrada 0. Ambas as transições a partir de s_3 são para ele mesmo.

31.



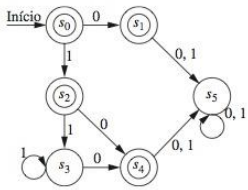
33. Estado inicial s_0, único estado final s_1; transições a partir de s_0 para s_0 com entrada 1, a partir de s_0 para s_1 com entrada 0, a partir de s_1 para s_1 com entrada 1; a partir de s_1 para s_0 com entrada 0.

35.



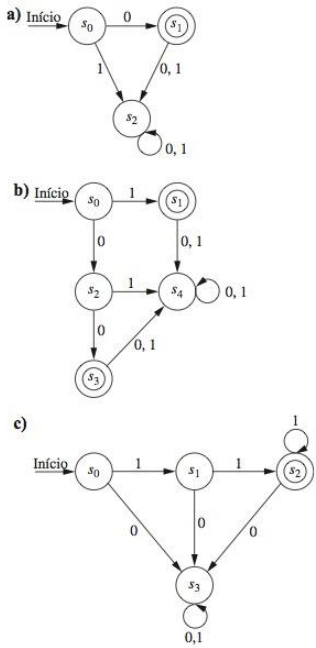
37. Suponha que tal máquina exista, com estado inicial s_0 e outro estado s_1. Como a seqüência vazia não está na linguagem, mas algumas seqüências são aceitas, devemos ter s_1 como único estado final, com pelo menos uma transição a partir de s_0 para s_1. Como a seqüência 0 não está na linguagem, a transição a partir de s_0 com entrada 0 deve ser para si mesmo, de modo que a transição a partir de s_0 com entrada 1 deve ser para s_1. Mas isto contradiz o fato de que 1 não está na linguagem. **39.** Mude cada estado final para um estado não final e vice-versa. **41.** Mesma máquina que no Exercício 25, mas com s_0, s_1 e s_2 como os estados finais. **43.** $\{0, 01, 11\}$ **45.** $\{\lambda, 0\} \cup \{0^m1^n \mid m \geq 1, n \geq 1\}$ **47.** $\{10^n \mid n \geq 0\} \cup \{10^n10^m \mid n, m \geq 0\}$ **49.** A união do conjunto de todas as seqüências que começam com um 0 e o conjunto de todas as seqüências que não têm nenhum 0.

51.



53. Adicione um estado não final s_3 com transições para s_3 a partir de s_0 com entrada 0, a partir de s_1 com entrada 1 e a partir de s_3 com entrada 0 ou 1.

55.



57. Suponha que M seja um autômato finito que aceite o conjunto das seqüências de bits que contêm números iguais de 0s e 1s. Suponha que M tenha n estados. Considere a seqüência $0^{n+1}1^{n+1}$. Pelo princípio da casa dos pombos, à medida que M processa esta seqüência, deve encontrar o mesmo estado mais de uma vez à medida que lê os primeiros $n + 1$ 0s; de modo que seja s um estado que ela atinge pelo menos duas vezes. Então, k 0s na entrada leva M a partir do estado s de volta para si mesmo para algum inteiro positivo k. Mas, então, M termina exatamente no mesmo lugar depois de ler $0^{n+1+k}1^{n+1}$ que termina depois de ler $0^{n+1}1^{n+1}$. Portanto, como M aceita $0^{n+1}1^{n+1}$, ele também aceita $0^{n+k+1}1^{n+1}$, o que é uma contradição. **59.** Sabemos, a partir do Exercício 58d, que as classes de equivalência de R_k são um refinamento das classes de equivalências de R_{k-1} para cada inteiro positivo k. As classes de equivalência são conjuntos finitos, e conjuntos finitos não podem ser refinados indefinidamente (o mais refinado que eles podem estar é cada classe de equivalência conter apenas um estado). Portanto, esta seqüência de refinamentos deve permanecer imutável a partir de certo ponto em diante. Resta mostrar que, assim que tivermos $R_n = R_{n+1}$, então $R_n = R_m$ para todo $m > n$, de onde segue que $R_n = R*$, e assim as classes de equivalência para estas duas relações serão as mesmas. Por indução, é suficiente mostrar que, se $R_n = R_{n+1}$, então $R_{n+1} = R_{n+2}$. Suponha que $R_{n+1} \neq R_{n+2}$. Isto significa que existem estados s

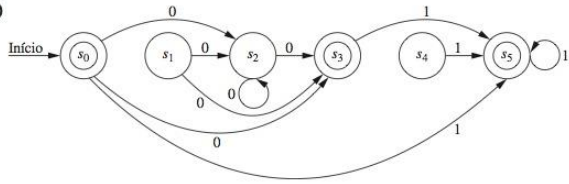
e t que são $(n + 1)$-equivalentes, mas não $(n + 2)$-equivalentes. Assim, existe uma seqüência x de comprimento $n + 2$ tal que, digamos, $f(s, x)$ é final, mas $f(t, x)$ não é final. Escreva $x = aw$, em que $a \in I$. Então, $f(s, a)$ e $f(t, a)$ não é $(n + 1)$-equivalente, pois w leva o primeiro para um estado final e o segundo para um estado não final. Mas $f(s, a)$ e $f(t, a)$ são n-equivalentes, pois s e t são $(n + 1)$-equivalentes. Isto contradiz o fato de que $R_n = R_{n+1}$. **61. a)** Pela maneira que a máquina $\overline{M}$ foi construída, uma seqüência levará M a partir do estado inicial para um estado final se e somente se aquela seqüência levar $\overline{M}$ a partir do estado inicial para um estado final. **b)** Para uma demonstração deste teorema, veja uma fonte como *Introduction to Automata Theory, Languages, and Computation* (2ª Edição), de John E. Hopcroft, Rajeev Motwani e Jeffrey D. Ullman (Addison-Wesley, 2000).

Seção 12.4

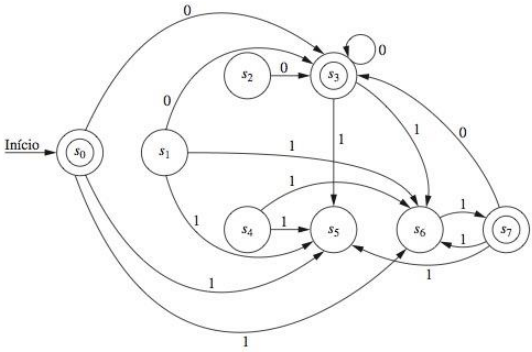
1. a) Qualquer número de 1s seguido por um 0. **b)** Qualquer número de 1s seguido por um ou mais 0s. **c)** 111 ou 001. **d)** Uma seqüência de qualquer número de 1s ou 00s ou alguns de cada em uma linha. **e)** λ ou uma seqüência que termina com um 1 e tem um ou mais 0s antes de cada 1. **f)** Uma seqüência de comprimento pelo menos 3 que termina com 00. **3. a)** Não. **b)** Não. **c)** Sim. **d)** Sim. **e)** Sim. **f)** Não. **g)** Não. **h)** Sim. **5. a) 0 ∪ 11 ∪ 010 b) 000000* c) (0 ∪ 1)((0 ∪ 1)(0 ∪ 1))* d) 0*10* e) (1∪01∪001)* 7. a) 00*1 b) (0 ∪ 1) (0 ∪ 1)(0 ∪ 1)*0000* c) 0*1* ∪ 1*0* d) 11(111)*(00)* 9. a)** Tem o estado inicial s_0, não final, sem nenhuma transição. **b)** Tem o estado inicial s_0, final, sem nenhuma transição. **c)** Tem o estado inicial não final s_0 e um estado final s_1 e a transição a partir de s_0 para s_1 com entrada a. **11.** Use uma demonstração por indução. Se a expressão regular para A for Ø, λ ou x, o resultado é trivial. Caso contrário, suponha que a expressão regular para A seja **BC**. Então, $A = BC$, em que B é o conjunto gerado por **B** e C é o conjunto gerado por **C**. Pela hipótese de indução, existem expressões regulares **B′** e **C′** que geram B^R e C^R, respectivamente. Como $A^R = (BC)^R = C^R B^R$, **C′B′** é uma expressão regular para A^R. Se a expressão regular para A for **B** ∪ **C**, então a expressão regular para A será **B′** ∪ **C′**, pois $(B \cup C)^R = (B^R) \cup (C^R)$. Finalmente, se a expressão regular para A for **B***, então é fácil ver que **(B′)*** é uma expressão regular para A^R.

13.

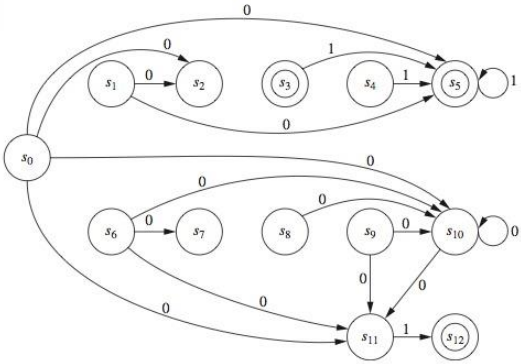
a)



b)



c)



15. $S \to 0A, S \to 1B, S \to 0, A \to 0B, A \to 1B, B \to 0B, B \to 1B$ **17.** $S \to 0C, S \to 1A, S \to 1, A \to 1A, A \to 0C, A \to 1, B \to 0B, B \to 1B, B \to 0, B \to 1, C \to 0C, C \to 1B, C \to 1$. **19.** Isto segue porque entradas que levam para um estado final no autômato correspondem univocamente a uma derivação na gramática. **21.** A parte "somente se" está clara, pois I é finito. Para a parte "se", sejam $s_{i_0}, s_{i_1}, s_{i_2}, ..., s_{i_n}$ os estados, em que $n = l(x)$. Como $n \geq |S|$, pelo princípio da casa dos pombos, algum estado está repetido. Seja y a parte de x que causa o laço, de modo que $x = uyv$ e y manda s_j para s_j, para algum j. Então $uy^kv \in L(M)$ para todo k. Logo, $L(M)$ é infinito. **23.** Suponha que $L = \{0^{2n}1^n\}$ fosse regular. Seja S o conjunto de estados de uma máquina de estado finito que reconheça este conjunto. Seja $z = 0^{2n}1^n$, em que $3n \geq |S|$. Então, pelo lema do bombeamento, $z = 0^{2n}1^n = uvw$, $l(v) \geq 1$ e $uv^iw \in \{0^{2n}1^n \mid n \geq 0\}$. Obviamente, v não pode conter ambos, 0 e 1, pois v^2 conteria então 10. Assim, v é todo 0s ou todo 1s, e, assim, uv^2w contém 0s demais ou 1s demais, de modo que ele não está em L. Esta contradição mostra que L não é regular. **25.** Suponha que o conjunto dos palíndromos em $\{0, 1\}$ fosse regular. Seja S o conjunto de estados de uma máquina de estado finito que reconheça este conjunto. Seja $z = 0^n10^n$, em que $n > |S|$. Aplique o lema do bombeamento para obter $uv^i w \in L$ para todo inteiro não negativo i em que $l(v) \geq 1$, e $l(uv) \leq |S|$, e $z = 0^n10^n = uvw$. Então, v deve ser uma seqüência de 0s (pois $|n| > |S|$), de modo que uv^2w não é um palíndromo. Logo, o conjunto dos palíndromos não é regular. **27.** Seja $z = 1$; então, $111 \notin L$, *mas* $101 \in L$, de modo que 11 e 10 são distinguíveis. Para a segunda pergunta, a única maneira de 1z estar em L é z terminar com 01, e esta também é a única maneira de 11z estar em L, de modo que 1 e 11 são indistinguíveis. **29.** Isto segue imediatamente do Exercício 28, pois as n seqüências distinguíveis devem levar a máquina do estado inicial para n estados diferentes. **31.** Quaisquer duas seqüências distintas de mesmo comprimento são distinguíveis em relação à linguagem P de todos os palíndromos, pois, se x e y forem seqüências distintas de comprimento n, então $xx^R \in P$, mas $yx^R \notin P$. Como existem 2^n seqüências diferentes de comprimento n, o Exercício 29 nos diz que qualquer autômato finito determinístico para reconhecer palíndromos deve ter pelo menos 2^n estados. Como n é arbitrário, isto é impossível.

Seção 12.5

1. a) A parte não branca da fita contém a seqüência 1111 quando a máquina pára. **b)** A parte não branca da fita contém a seqüência 011 quando a máquina pára. **c)** A parte não branca da fita contém a seqüência 00001 quando a máquina pára. **d)** A parte não branca da fita contém a seqüência 00 quando a máquina pára. **3. a)** A máquina pára (e aceita) em um branco seguindo a entrada, tendo mudado a fita de 11 para 01. **b)** A máquina muda uma a cada duas ocorrências de um 1, se ocorrer algum, começando com o primeiro, para um 0, e, caso contrário, não muda a seqüência; ela pára (e aceita) quando chegar ao final da seqüência. **5. a)** Pára com 01 na fita, e não aceita. **b)** O primeiro 1 (se houver algum) é mudado para um 0 e os outros não são mudados. A entrada não é aceita.

7. $(s_0, 0, s_1, 1, R), (s_0, 1, s_0, 1, R)$ **9.** $(s_0, 0, s_0, 0, R), (s_0, 1, s_1, 1, R), (s_1, 0, s_1, 0, R), (s_1, 1, s_1, 0, R)$ **11.** $(s_0, 0, s_1, 0, R), (s_0, 1, s_0, 0, R), (s_1, 0, s_1, 0, R), (s_1, 1, s_0, 0, R), (s_1, B, s_2, B, R)$ **13.** $(s_0, 0, s_0, 0, R), (s_0, 1, s_1, 1, R), (s_1, 0, s_1, 0, R), (s_1, 1, s_0, 1, R), (s_0, B, s_2, B, R)$ **15.** Se a seqüência de entrada for branca ou começar com um 1, a máquina pára no estado não final s_0. Caso contrário, o 0 inicial é mudado para um M e a máquina pula todos os 0s e 1s intermediários até que ela ou chegue ao final da seqüência entrada ou vá para um M. Neste ponto, ela volta um quadrado e entra no estado s_2. Como as seqüências aceitáveis devem ter um 1 na direita para cada 0 na esquerda, deve haver um 1 aqui se a seqüência for aceitável. Portanto, a única transição saindo de s_2 ocorre quando este quadrado contém um 1. Se contiver, a máquina o substitui por um M e volta para a esquerda; se não contiver, a máquina pára no estado não final s_2. Em seu caminho de volta, permanece em s_3 enquanto

encontrar 1s, então permanece em s_4 enquanto encontrar 0s. Eventualmente, ou ela encontra um 1 enquanto em s_4, em cujo ponto pára sem aceitar, ou então atinge a parte mais à direita de M que foi escrita sobre um 0 no início da seqüência. Se ela estiver em s_3 quando isto ocorrer, então não existem mais 0s na seqüência, de modo que seria melhor que também não houvesse mais nenhum 1; isto pode ser alcançado com as transições (s_3, M, s_5, M, R) e (s_5, M, s_6, M, R), e s_6 é um estado final. Caso contrário, a máquina pára em um estado não final s_5. Se estiver em s_4 quando este M for encontrado, começa tudo de novo, exceto que agora a seqüência terá seu 0 restante mais à esquerda e seu 1 restante mais à direita substituídos por Ms. Assim, a máquina se move, permanecendo no estado s_4, para o 0 restante mais à esquerda e volta para o estado s_0 para repetir o processo.

17. (s_0, B, s_9, B, L), $(s_0, 0, s_1, 0, L)$, (s_1, B, s_2, E, R), (s_2, M, s_2, M, R), $(s_2, 0, s_3, M, R)$, $(s_3, 0, s_3, 0, R)$, (s_3, M, s_3, M, R), $(s_3, 1, s_4, M, R)$, $(s_4, 1, s_4, 1, R)$, (s_4, M, s_4, M, R), $(s_4, 2, s_5, M, R)$, $(s_5, 2, s_5, 2, R)$, (s_5, B, s_6, B, L), (s_6, M, s_8, M, L), $(s_6, 2, s_7, 2, L)$, $(s_7, 0, s_7, 0, L)$, $(s_7, 1, s_7, 1, L)$, $(s_7, 2, s_7, 2, L)$, (s_7, M, s_7, M, L), (s_7, E, s_2, E, R), (s_8, M, s_8, M, L), (s_8, E, s_9, E, L), em que M e E são marcadores, com E marcando a extremidade esquerda da entrada

19. $(s_0, 1, s_1, B, R)$, $(s_1, 1, s_2, B, R)$, $(s_2, 1, s_3, B, R)$, $(s_3, 1, s_4, 1, R)$, $(s_1, B, s_4, 1, R)$, $(s_2, B, s_4, 1, R)$, $(s_3, B, s_4, 1, R)$

21. $(s_0, 1, s_1, B, R)$, $(s_1, 1, s_2, B, R)$, (s_1, B, s_6, B, R), $(s_2, 1, s_3, B, R)$, (s_2, B, s_6, B, R), $(s_3, 1, s_4, B, R)$, (s_3, B, s_6, B, R), $(s_4, 1, s_5, B, R)$, (s_4, B, s_6, B, R), $(s_6, B, s_{10}, 1, R)$, $(s_5, 1, s_5, B, R)$, $(s_5, B, s_7, 1, R)$, $(s_7, B, s_8, 1, R)$, $(s_8, B, s_9, 1, R)$, $(s_9, B, s_{10}, 1, R)$

23. $(s_0, 1, s_0, 1, R)$, (s_0, B, s_1, B, L), $(s_1, 1, s_2, 0, L)$, $(s_2, 0, s_2, 0, L)$, $(s_2, 1, s_3, 0, R)$, (s_2, B, s_6, B, R), $(s_3, 0, s_3, 0, R)$, $(s_3, 1, s_3, 1, R)$, $(s_3, B, s_4, 1, R)$, $(s_4, B, s_5, 1, L)$, $(s_5, 1, s_5, 1, L)$, $(s_5, 0, s_2, 0, L)$, $(s_6, 0, s_6, 1, R)$, $(s_6, 1, s_7, 1, R)$, (s_6, B, s_7, B, R)

25. $(s_0, 0, s_0, 0, R)$, $(s_0, *, s_5, B, R)$, $(s_3, *, s_3, *, L)$, $(s_3, 0, s_3, 0, L)$, $(s_3, 1, s_3, 1, L)$, (s_3, B, s_0, B, R), $(s_5, 1, s_5, B, R)$, $(s_5, 0, s_5, B, R)$, (s_5, B, s_6, B, L), (s_6, B, s_6, B, L), $(s_6, 0, s_7, 1, L)$, $(s_7, 0, s_7, 1, L)$, $(s_0, 1, s_1, 0, R)$, $(s_1, 1, s_1, 1, R)$, $(s_1, *, s_2, *, R)$, $(s_2, 0, s_2, 0, R)$, $(s_2, 1, s_3, 0, L)$, (s_2, B, s_4, B, L), $(s_4, 0, s_4, 1, L)$, $(s_4, *, s_8, B, L)$, $(s_8, 0, s_8, B, L)$, $(s_8, 1, s_8, B, L)$

27. Suponha que s_m seja o único estado de parada para a máquina de Turing no Exercício 22, em que m é o maior número de estado, e suponha que tenhamos projetado a máquina de modo que, quando a máquina pára a fita, está lendo o 1 mais à esquerda da resposta. Renumere cada estado desta máquina para o Exercício 18 adicionando m para cada subscrito, e considere a união de dois conjuntos de cinco-uplas.

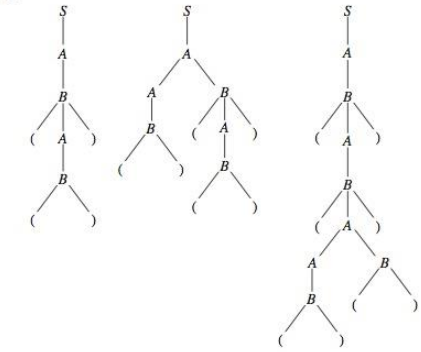
29. **a)** Não. **b)** Sim. **c)** Sim. **d)** Sim.

31. $(s_0, B, s_1, 1, L)$, $(s_0, 1, s_1, 1, R)$, $(s_1, B, s_0, 1, R)$

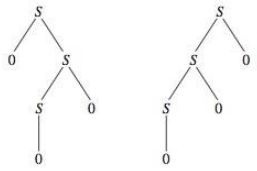
Exercícios Complementares

1. **a)** $S \to 00S111$, $S \to \lambda$ **b)** $S \to AABS$, $AB \to BA$, $BA \to AB$, $A \to 0$, $B \to 1$, $S \to \lambda$ **c)** $S \to ET$, $T \to 0TA$, $T \to 1TB$, $T \to \lambda$, $0A \to A0$, $1A \to A1$, $0B \to B0$, $1B \to B1$, $EA \to E0$, $EB \to E1$, $E \to \lambda$

3.

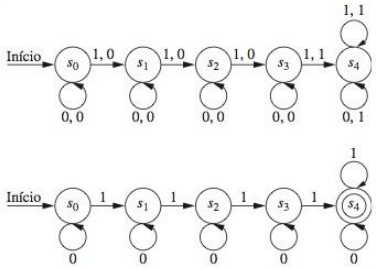


5.

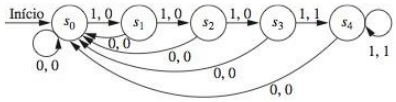


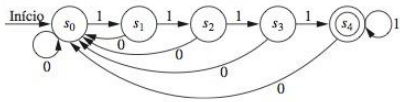
7. Não, considere $A = \{1, 10\}$ e $B = \{0, 00\}$. **9.** Não, considere $A = \{00, 000, 00000\}$ e $B = \{00, 000\}$. **11.** **a)** 1 **b)** 1 **c)** 2 **d)** 3 **e)** 2 4

13.



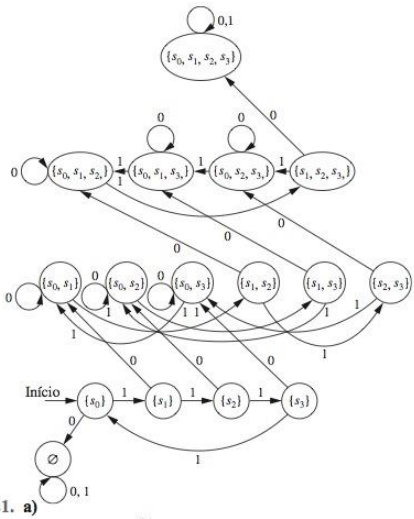
15.



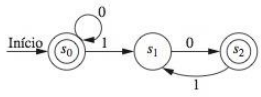


17. **a)** $n^{nk+1}m^{nk}$ **b)** $n^{nk+1}m^{n}$

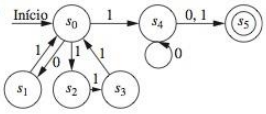
19.



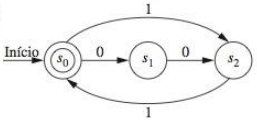
21. **a)**



b)



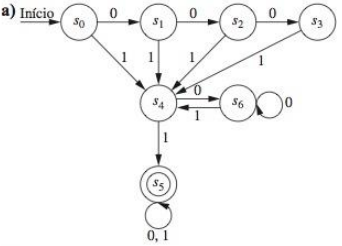
c)



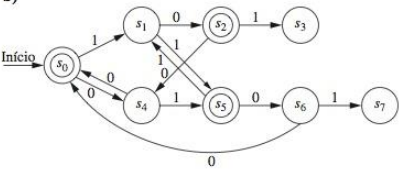
23. Construa o autômato finito determinístico para A com estados S e estados finais F. Para $\overline{A}$ use o mesmo autômato, mas com estados finais $S - F$.

25.

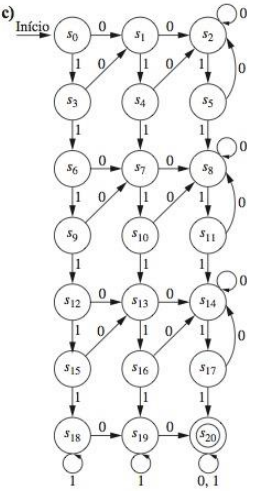
a)



b)



c)



27. Suponha que $L = \{1^p \mid p \text{ é primo}\}$ seja regular, e seja S o conjunto dos estados em um autômato finito que reconheça L. Seja $z = 1^p$, em que p é um primo com $p > |S|$ (tal primo existe, pois existem infinitos primos). Pelo lema do bombeamento, deve ser possível escrever $z = uvw$ com $l(uv) \leq |S|$, $l(v) \geq 1$, e para todos os inteiros não negativos i, $uv^iw \in L$. Como z é uma seqüência só de 1s, $u = 1^a$, $v = 1^b$ e $w = 1^c$, em que $a + b + c = p$, $a + b \leq n$ e $b \geq 1$. Isto significa que $uv^iw = 1^a1^{bi}1^c = 1^{(a+b+c)+b(i-1)} = 1^{p+b(i-1)}$. Agora, considere $i = p + 1$. Então, $uv^iw = 1^{p(1+b)}$. Como $p(1 + b)$ não é primo, $uv^iw \notin L$, o que é uma contradição. **29.** $(s_0, *, s_5, B, L)$, $(s_0, 0, s_0, 0, R)$, $(s_0, 1, s_1, 0, R)$, $(s_1, *, s_2, *, R)$, $(s_1, 1, s_1, 1, R)$, $(s_2, 0, s_2, 0, R)$, $(s_2, 1, s_3, 0, L)$, (s_2, B, s_4, B, L), $(s_3, *, s_3, *, L)$, $(s_3, 0, s_3, 0, L)$, $(s_3, 1, s_3, 1, L)$, (s_3, B, s_0, B, R), $(s_4, *, s_8, B, L)$, $(s_4, 0, s_4, B, L)$, $(s_5, 0, s_5, B, L)$, (s_5, B, s_6, B, R), $(s_6, 0, s_7, 1, R)$, (s_6, B, s_6, B, R), $(s_7, 0, s_7, 1, R)$, $(s_7, 1, s_7, 1, R)$, $(s_8, 0, s_8, 1, L)$, $(s_8, 1, s_8, 1, L)$

APÊNDICES

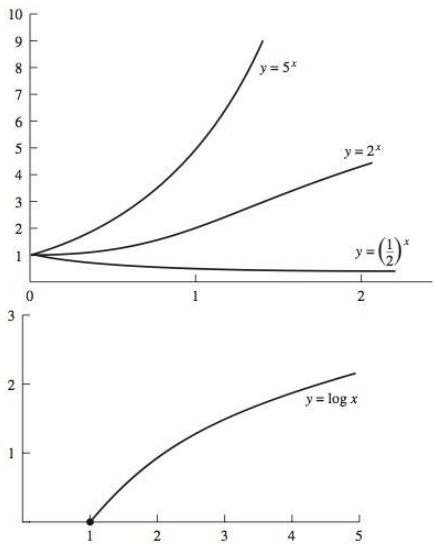
Apêndice 1

1. Suponha que $1'$ também seja um elemento neutro da multiplicação de números reais. Então, por definição, temos ambos, $1 \cdot 1' = 1$ e $1 \cdot 1' = 1'$, de modo que $1' = 1$. **3.** Para a primeira parte, basta mostrar que $[(-x) \cdot y] + (x \cdot y) = 0$, pois o Teorema 2 garante que os inversos aditivos são únicos. Logo, $[(-x) \cdot y] + (x \cdot y) = (-x + x) \cdot y$ (pela propriedade distributiva) $= 0 \cdot y$ (pela propriedade dos elementos inversos) $= y \cdot 0$ (pela propriedade comutativa) $= 0$ (pelo Teorema 5). A segunda parte é quase idêntica. **5.** Basta mostrar que $[(-x) \cdot (-y)] + [-(x \cdot y)] = 0$, pois o Teorema 2 garante que os inversos aditivos são únicos: $[(-x) \cdot (-y)] + [-(x \cdot y)] = [(-x) \cdot (-y)] + [(-x) \cdot y]$ (pelo Exercício 3) $= (-x) \cdot [(-y) + y]$ (pela propriedade distributiva) $= (-x) \cdot 0$ (pela propriedade dos elementos inversos) $= 0$ (pelo Teorema 5). **7.** Por definição, $-(-x)$ é o inverso aditivo de $-x$. Mas $-x$ é o inverso aditivo de x, de modo que x é o inverso aditivo de $-x$. Portanto, $-(-x) = x$ pelo Teorema 2. **9.** Basta mostrar que $(-x - y) + (x + y) = 0$, pois o Teorema 2 garante que os inversos aditivos são únicos: $(-x - y) + (x + y) = [(-x) + (-y)] + (x + y)$ (pela definição de subtração) $= [(-y) + (-x)] + (x + y)$ (pela propriedade comutativa) $= (-y) + [(-x) + (x + y)]$ (pela propriedade associativa) $= (-y) + [(-x + x) + y]$ (pela propriedade associativa) $= (-y) + (0 + y)$ (pela propriedade dos elementos inversos) $= (-y) + y$ (pela propriedade dos elementos neutros) $= 0$ (pela propriedade dos elementos inversos). **11.** Por definição de divisão e pela unicidade dos inversos multiplicativos (Teorema 4), basta demonstrar que $[(w/x) + (y/z)] \cdot (x \cdot z) = w \cdot z + x \cdot y$. Mas isto segue depois de diversos passos, usando as propriedades distributiva, associativa e comutativa para a multiplicação e a definição de a divisão ser o mesmo que a multiplicação pelo inverso. **13.** Devemos mostrar que, se $x > 0$ e $y > 0$, então $x \cdot y > 0$. Pela propriedade da compatibilidade da multiplicação, pela propriedade comutativa e pelo Teorema 5, temos $x \cdot y > 0 \cdot y = 0$. **15.** Observe primeiro que, se $z < 0$, então $-z > 0$ (some $-z$ a ambos os lados da hipótese). Agora, dados $x > y$ e $-z > 0$, temos $x \cdot (-z) > y \cdot (-z)$ pela propriedade da compatibilidade da multiplicação. Mas, pelo Exercício 3, isto é equivalente a $-(x \cdot z) > -(y \cdot z)$. Então, some $x \cdot z$ e $y \cdot z$ a ambos os lados e aplique as diversas propriedades nas maneiras óbvias para obter $x \cdot z < y \cdot z$. **17.** A propriedade da compatibilidade da adição nos diz que $w + y < x + y$ e (junto com a propriedade comutativa) que $x + y < x + z$. Pela propriedade transitiva, isto fornece a conclusão desejada. **19.** Pelo Teorema 8, aplicado a $1/x$ no lugar de x, existe um inteiro n (necessariamente positivo, pois $1/x$ é positivo) tal que $n > 1/x$. Pela propriedade da compatibilidade da multiplicação, isto significa que $n \cdot x > 1$. **21.** Devemos mostrar que, se $(w, x) \sim (w', x')$ e $(y, z) \sim (y', z')$, então $(w + y, x + z) \sim (w' + y', x' + z')$ e que $(w \cdot y + x \cdot z, x \cdot y + w \cdot z) \sim (w' \cdot y' + x' \cdot z', x' \cdot y' + w' \cdot z')$. Assim, nos foi dado que $w + x' = x + w'$ e que $y + z' = z + y'$, e queremos mostrar que $w + y + x' + z' = x + z + w' + y'$ e que $w \cdot y + x \cdot z + x' \cdot y' + w' \cdot z' = x \cdot y + w \cdot z + w' \cdot y' + x' \cdot z'$. Para a primeira das conclusões desejadas, some as duas equações dadas. Para a segunda, reescreva as equações dadas como $w - x = w' - x'$ e $y - z = y' - z'$, multiplique-as e faça as contas.

Apêndice 2

1. **a)** 2^3 **b)** 2^6 **c)** 2^4 **3.** **a)** $2y$ **b)** $2y/3$ **c)** $y/2$

5.



Apêndice 3

1. Depois de o primeiro bloco ter sido executado, a *a* foi associado o valor original de *b* e a *b* foi associado o valor original de *c*, enquanto, depois de o segundo bloco ter sido executado, a *b* é associado o valor original de *c* e a *a*, o valor original de *c* também. **3.** A seguinte construção **while** faz a mesma coisa.

i := *valor inicial*
while *i* ≤ *valor final*
begin
  comando
  i := *i* + 1
end

Créditos das Fotos

CAPÍTULO 1

Página 2: © National Library of Medicine; **p. 5:** Library of Congress; **p. 14:** Cortesia de Indiana University Archives; **p. 16:** University Archives. Department of Rare Books and Special Collections. Princeton University Library; **p. 25:** Com permissão da London Mathematical Society; **p. 27:** © Mary Evans Picture Library; **p. 28:** Henry Maurice Sheffer (1883–1964), de Mabel Lisle Ducasse. Cortesia de Harvard University Portrait Collection, Department of Philosophy, CNA1. Foto de Katya Kallsen; **p. 32:** © Bettmann/Corbis Images; **p. 44:** © Granger Collection; **p. 93:** © Master and Fellows of Trinity College, Cambridge; **p. 94:** © Master and Fellows of Trinity College Cambridge.

CAPÍTULO 2

Página 113: Library of Congress; **p. 114:** Library of Congress; **p. 115:** Com permissão do presidente e do conselho da Royal Society; **p. 117:** © Scientists and Inventors Portrait File Collection, Archives Center, National Museum of American History; **p. 154:** Cortesia de Neil Sloane e do AT&T Shannon Lab.

CAPÍTULO 3

Página 168: © In THE SCIENTISTS OF THE ANCIENT WORLD, de Anderson/Stephenson. p. 90. Ilustração de Marie le Glatin Keis; **p. 182:** Bachmann, Paul. Die Arithmetik Der Quadratischen Formen. Verlag und Druck von BG Teubner. Leipzig. Berlim, 1923; **p. 183:** Smith Collection, Rare Book and Manuscript Library, Columbia University; **p. 184:** © Stanford University News Service; **p. 204:** Smithsonian Institution. Foto nº 46, 834-M; **p. 213:** © Mary Evans Picture Library; **p. 228:** Scientists and Inventors Portrait File Collection, Archives Center, National Museum of American History; **p. 239:** In *A History of Science, Technology and Philosophy* in the 16th and 17th Centuries, de Abraham Wolf. Reimpresso com permissão de Harper Collins Publishers Lts. © Abraham Wolf; **p. 242, superior:** Cortesia de Ronald L. Rivest; **p. 242, centro:** © Kristen Tsolis; **p. 242, inferior:** © Cortesia de Leonard Adleman.

CAPÍTULO 4

Página 298: © Granger Collection; **p. 299:** © Mary Evans Picture Library; **p. 324:** Cortesia de Tony Hoare; **p. 332:** © Matthew Naythons/TimePix/Getty.

CAPÍTULO 5

Página 348: © Granger Collection; **p. 352:** Cortesia de Mrs. Jane Burch; **p. 366:** © National Library of Medicine.

CAPÍTULO 6

Página 394: In *A Concise History of Mathematics,* de Dirk Struik, 1967, Dover Publications; **p. 407:** © Granger Collection; **p. 420:** Cortesia de Stephen Stigler; **p. 438:** © Smith Collection, Rare Book and Manuscript Library, Columbia University.

CAPÍTULO 7

Página 508: In Math Tutor Archive.

CAPÍTULO 8

Página 551: Cortesia de Stephen Warshall; **p. 571:** © Deutsches Museum; **p. 584:** Com permissão do presidente e do conselho da Royal Society.

CAPÍTULO 9

Página 635: © Scientists and Inventors Portrait File Collection, Archives Center, National Museum of American History; **p. 640:** Master and Fellows of Trinity College Cambridge; **p. 646:** Julius Petersen: Metoder og Teorier. Edition nº 6, Det Schoenbergske Forlag, Copenhagen, 1912. Foto: cortesia de Bjarne Toft; **p. 650:** Cortesia de Edsger W. Dijkstra; **p. 663:** Cortesia de Archive of Polish Mathematicians, do Institute of Mathematics of the Polish Academy of Sciences. Reproduzido com permissão do instituto; **p. 668:** Com permissão da London Mathematical Society.

CAPÍTULO 10

Página 688: © Master and Fellows of Trinity College Cambridge; **p. 702:** Cortesia da California State University, Santa Cruz; **p. 721:** Cortesia do Archive of Polish Mathematicians, do Institute of Mathematics of the Polish Academy of Sciences. Reproduzido com permissão do instituto; **p. 738:** Cortesia de Robert Clay Prim; **p. 739:** Cortesia de Joseph B. Kruskal. Foto de Rachel Kruskal.

CAPÍTULO 11

Página 750: Cortesia do MIT Museum; **p. 768:** Cortesia de Maurice Karnaugh; **p. 775:** Cortesia de Edward J. McCluskey; **p. 776:** Cortesia da Quine Family.

CAPÍTULO 12

Página 791: © Donna Coveney/MIT; **p. 792, superior:** Cortesia IBM, Almaden Research Center; **p. 792, inferior:** Cortesia de Peter Naur; **p. 805:** In: *I Have a Photographic Memory,* de Paul Halmos (1987), American Mathematical Society; **p. 811:** © Bettmann/Corbis; **p. 826:** © Science Photo Library/Photo Researchers, Inc. **p. 836:** © University Archives. Department of Rare Books and Special Collections. Princeton University Library.

APÊNDICE:

Página 845, Apêndice 1: © Hulton-Deutsch Collection/Corbis.

Índice de Nomes

Ada, Augusta (Condessa de Lovelace), 27
Adleman, Leonard, 242
Archimedes, 848
Aristóteles, 2

Bachmann, Paul Gustav Heinrich, 182
Backus, John, 792
Bayes, Thomas, 420
Bernoulli, James, 407
Boole, George, 5

Cantor, Georg, 113
Carmichael, Robert Daniel, 240
Carroll, Lewis (Charles Dodgson), 44
Cayley, Arthur, 688
Chebyshev, Pafnuty Lvovich, 438
Chomsky, Avram Noam, 791
Church, Alonzo, 836

De Morgan, Augustus, 25
Descartes, René, 117
Dijkstra, Edsger Wybe, 650
Dirichlet, G. Lejeune, 348
Dodgson, Charles (Lewis Carroll), 44

Eratóstenes, 508
Erdös, Paul, 584
Euclides, 228
Euler, Leonhard, 214, 635

Fermat, Pierre de, 239
Fibonacci, 298

Gauss, Karl Friedrich, 204
Goldbach, Christian, 215

Hamilton, William Rowan, 640
Hardy, Godfrey Harold, 93
Hasse, Helmut, 571
Hoare, C. Anthony R., 324
Hopper, Grace Brewster Murray, 811
Huffman, David A., 702

Karnaugh, Maurice, 768
Kempe, Alfred Bray, 668
al-Khowarizmi, Abu Ja'far Mohammed ibn Musa, 168
Kleene, Stephen Cole, 805 Knuth, Donald E., 184
Kruskal, Joseph Bernard, 739
Kuratowski, Kazimierz, 663

Lamé, Gabriel, 299
Landau, Edmund, 183
Laplace, Pierre-Simon, 394
Lukasiewicz, Jan, 721

McCarthy, John, 332
McCluskey, Edward J., 775
Mersenne, Marin, 213

Naur, Peter, 792

Pascal, Blaise, 366
Peirce, Charles Sanders, 32
Petersen, Julius Peter Christian, 646
Prim, Robert Clay, 738

Quine, Willard van Orman, 776

Ramanujan, Srinivasa, 94
Ramsey, Frank Plumpton, 352
Rivest, Ronald, 242
Russell, Bertrand, 114

Shamir, Adi, 242
Shannon, Claude Elwood, 750
Sheffer, Henry Maurice, 28
Sloane, Neil, 154
Smullyan, Raymond, 14
Stirling, James, 145

Tukey, John Wilder, 16
Turing, Alan Mathison, 176, 826

Vandermonde, Alexandre-Théophile, 368
Venn, John, 115

Warshall, Stephen, 551

Índice Remissivo

A menos, 6
Abordagem
  bottom-up parsing, 791
  top-down parsing, 791
Academia de Platão, 2
Aceitação, algoritmo de, 179
Ackermann, Wilhelm, 310
Acréscimo, 206
Ada, Augusta (Condessa de Lovelace), 25, 27
Adaptando demonstrações de existência, 96
Adição
  de funções, 135–136
  de inteiros, 222–224
  de matrizes, 247–248
  de multiconjuntos, 132
  de progressões geométricas, 270–271
  de subconjuntos, 732
Adjetivo, 786
Adleman, Leonard, 241, 242
Advérbio, 786
Agrimensura da Costa dos Estados Unidos, 32
Alça (handle), 745
Alcuin de York, 632
Alef zero, 158
Alexandre, o Grande, 2
Alfabeto, 787, 838
Alfabeto de entrada, 785, 798, 805
Alfabeto de saída, 798
Álgebra booleana, 5, 28, 752–755
  definição abstrata de, 755
  definição de, 781
  identidades da, 752–754, 755
ALGOL, 792
ALGOL 60, 792
Algoritmo de aceitação, 179, 293
Algoritmo de aproximação, 655
Algoritmo de busca binária, 171, 257, 474
  complexidade de, 194–195, 197
  recursivo, 314
Algoritmo de busca linear, 170, 257
  complexidade de, 194, 197
  complexidade de médio caso de, 430–431
  recursivo, 314
Algoritmo de busca ternária, 178
Algoritmo de Dijkstra, 649–653, 676
Algoritmo de divisão, 202–203
Algoritmo de Euclides, 227–229, 258, 298
  estendido, 232, 246
Algoritmo de Euclides estendido, 232, 246
Algoritmo de Fleury, 637, 646
Algoritmo de Floyd, 656–657
Algoritmo de Hasse, 101–102
Algoritmo de iteração, para números Fibonacci, 317
Algoritmo de Kruskal, 739–740, 744
Algoritmo de Prim, 738–739, 744
Algoritmo de Roy-Warshall, 551–553
Algoritmo de Sollin, 743
Algoritmo de Warshall, 550–553
Algoritmo do menor caminho, 649–653
Algoritmo ótimo, 200
Algoritmo voraz, 174–176
Algoritmo voraz, 174–176, 275–276, 703, 738, 744
Algoritmo voraz para organizar trocos, 175
Algoritmo(s), 167–177
  aceitação, 179, 293
  aproximação, 655
  bubble sort, 173
  busca, 170, 257
    binária, 171, 257, 474
    booleana, 13
    complexidade de, 193–195, 197
    em largura, 729–731, 744
    em profundidade, 726–729, 744
      aplicações com, 727–729
      em grafos orientados, 732–734
    linear, 170, 256, 257
      complexidade de, 194, 197
      complexidade caso de médio de, 431–432
      seqüencial recursiva, 314
    ternária, 178
    Web page, 13
  complexidade, 193–199
    computacional, 193
      do algoritmo de Dijkstra, 653
    constante, 197
    de algoritmos de busca em largura, 729
    de caso médio, 195, 431–433
    de merge sort, 479
    do algoritmo de busca binária, 194–195
    do algoritmo de busca em largura, 731
    do algoritmo de busca linear, 194
    do bubble sort, 194–195
    do insertion sort, 196
    dos algoritmos de busca, 194–195, 197
    dos algoritmos de ordenação, 195–196
    espacial, 193
    exponencial, 197
    fatorial, 197
    linear, 197
    logarítmica, 197
    para encontrar o maior elemento de um conjunto, 194
    pior caso, 194, 195–196
    polinomial, 197
    temporal, 193–196
  de caminho mais curto, 649–653
  de Euclides, 227–229, 258, 298
    estendido, 232, 246
  de Fleury, 637, 646
  de Floyd, 656–657
  de Kruskal, 739–740, 744
  de percurso, 712–719
  de Prim, 738–739, 744
  de Sollin, 743
  de Warshall, 550–553, 582
  definição de, 168
  Dijkstra, 649–653, 676
  divisão, 202–203, 257
  divisão e resolução, 474, 514
  estimativa do número de operações de, 180–182
  gerando combinações, 384–385
  gerando permutações, 382–384
  história da palavra, 168
  Codificação de Huffman, 702–703
  insertion sort, 174
  inteiros e, 219–229
  iteração, para computar números de Fibonacci, 317
  merge sort, 317–321
  Monte Carlo, 411–413
  NAUTY, 618
  ordenação, 172–174
    complexidade, 195–196
    topológica, 576–578, 582
  ordenação topológica, 577
  ótimo, 200
  para adição de números inteiros, 222–224
  para árvores geradoras, 738–741
  para árvores geradoras mínimas, 738–741
  para avaliações polinomiais, 199
  para busca por árvore binária, 695–707
  para caminhos eulerianos, 637–638
  para ciclos eulerianos, 636–637
  para colorir grafos, 671, 674
  para computar div e mod, 225–226
  para encontrar o maior elemento de uma seqüência, 169
  para encontrar o menor elemento de uma seqüência, 259
  para expansão na base *b*, 219, 221
  para fecho transitivo, 550
  para multiplicação de inteiros, 224–226
  para multiplicação de matrizes, 249–250
  para multiplicação rápida de inteiros, 475–476
  para multiplicação rápida de matriz, 476
  para números de Fibonacci, 316–317
  para operações com inteiros, 222–226
  para potenciação modular, 226
  para produto booleano de matrizes zero-um, 252–254
  paralelo, 606
  percurso em pós-ordem, 718,

percurso em pré-ordem, 718
probabilístico, 393, 411–413, 442
propriedades de, 169
recursivos, 311–321, 329
busca linear, 314
correção de, 315–317
de busca binária, 314
para computar a^n, 312
para computar máximo divisor comum, 313
para fatoriais, 197
para números de Fibonacci, 316
potenciação modular, 313
seriais, 606
tempo de computação usado para, 198–199
voraz, 174–176, 275–276, 703, 738, 744
voraz para organizar trocos, 174–176
Algoritmo de busca seqüencial recursivo, 314
Algoritmos de busca, 170–172, 257
binária, 171–172, 257, 474
binária recursiva, 314
booleana, 13
complexidade de, 194–195, 197
largura, 729–731, 744
linear, 170, 257
complexidade de, 195, 197
complexidade de caso médio de, 431–432
recursiva, 314
linear recursiva, 314
profundidade, 726–729, 744
aplicações com, 732–734
em grafos orientados, 732–734
seqüencial recursiva, 314
ternária, 178
Web, 13
Algoritmos de ordenação, 172–174
complexidade de, 195–196
topológica, 576–578, 582
Algoritmos Monte Carlo, 411–413
Algoritmos paralelos, 606
Algoritmos probabilísticos, 393, 411–413, 442
Algoritmos recursivos, 311–321, 329
correção de, 315
de busca binária, 314
para busca linear, 314
para computar a^n, 312
para computar o máximo divisor comum, 313
para fatoriais, 311–312
para números Fibonacci, 316
para potenciação modular, 313
provar o correto, 315–317
seguimento de, 312, 313
Algoritmos seriais, 606
Alice no País das Maravilhas (Carroll), 44
al-Khowarizmi, Abu Ja'far Mohammed ibn Musa, 168
Altura da árvore binária completa, 306
Altura da árvore enraizada, 691, 743
Altura estrela, 840
Amostra
com reposição, 396
sem reposição, 396
Análise top-down, 791
Ancestrais de um vértice, 686, 743
Antecedente, 6
Anticadeia, 585
Aplicação(ões)
da inclusão-exclusão, 505–512
da teoria de grafos, 592–595, 604–607, 623–625, 627, 638, 643, 671–672
da teoria dos números, 231–244
de árvores, 695–707
de coloração de grafos, 671–672
de congruências, 205–207
do princípio da casa dos pombos, 351–353
do Teorema de Bayes, 419–421
em profundidade, 731–732
Apologia de um Matemático, *Uma* (Hardy), 93
Apostas, 393
Appel, Kenneth, 668, 669
Aresta de corte, 625
Aresta incidente, 598
Arestas
corte, 625
de grafo não orientado, 592, 600
de grafo orientado, 591, 600–601
de grafo simples, 590, 611
de multigrafo, 590
de multigrafo orientado, 591
de pseudografo, 591
extremidades de, 598
incidentes, 598, 676
indiretas, 592, 600, 675
múltiplas, 590, 591, 675
não orientadas, 592, 675
Arestas da árvore, 728
Arestas de retorno, 728
Arestas múltiplas, 590, 591, 592, 675
Arestas não orientadas, 592, 675
de grafo simples, 590
Arestas orientadas, 592, 600–601, 675
Arestas paralelas, 590
Argumento, 44, 63
diagonalização de Cantor, 160
forma, 64
válido, 64
Argumento de diagonalização, 160
Argumento de diagonalização de Cantor, 160
Argumento dedutivo, 264
Argumento indutivo, 264
Aristóteles, 2
Arithmeticorum Libri Duo (Maurolico), 265
Aritmética
da computação para números inteiros grandes, 237–238
de matrizes, 247–248
modular, 203–205
Teorema Fundamental da, 211, 258, 285–286
Aritmética da computação, para números inteiros grandes, 237–238
Aritmética modular, 203–205
Arquimedes, 848
Ars Conjectandi (Bernoulli), 407
Art of Computer Programming, The (Knuth), 172
Artigo, 786
Artigos acadêmicos, 593
Árvore(s), 683–741
altura balanceada, 748
aplicações de, 695–707
AVL, 748
binária, 302–304, 686, 687
completa, 304–305
estendida, 302
binária completa, 303
binária estendida, 302
binomial, 745
centopéia, 746
como modelos, 688–690
de busca binária, 695–698, 743
de decisão, 698–700, 743
de Fibonacci enraizada, 695
de jogo, 704–707
definição de, 683
derivação, 790–792, 839
enraizada (ou com raiz), 302, 685–688
altura de, 691, 743
árvores de decisão, 698–700
balanceada, 691, 743
B-árvore de grau k, 745
binomial, 745
definição de, 743
ordem de nível de vértices de, 745
ordenada, 687, 743, 745
S_k-árvore, 745
genealógica, 684
geradora, 724–734, 744
construção
por busca em largura, 729–731
por busca em profundidade, 726–729
de grau restrito, 746
definição de, 724
distância entre, 737
distribuição múltipla em IP, 726
enraizada, 737
máxima, 742
mínima, 737–741, 744
graciosa, 746
grau m, 686, 743
altura de, 692–693
cheia, 694
m-ária completa, 686, 690, 743
mesh de, 748
propriedades de, 690–693
quad, 748
rotulada, 695
Árvore de derivação (ou *parse tree*), 790–792, 839
Árvore de grau m, 686, 743
altura de, 692–693
cheia, 686, 690
completa, 694
Árvore de grau m cheia, 686, 690, 743
completa, 694
Árvore de grau m completa, 694
Árvore enraizada balanceada, 691, 743
Árvore enraizada ordenada, 687, 743, 745
Árvore genealógica de Bernoulli, 684

Árvore geradora de grau restrito, 746
Árvore geradora máxima, 742
Árvore organizacional, 689
Árvore rotulada, 695
Árvores AVL, 748
Árvores balanceadas por altura, 748
Árvores binárias, 302–304, 686, 687
  com códigos de prefixo, 700–701
  completas, 304–305
    altura de, 306
    número de vértices de, 306
  definição de, 743
  estendidas, 302
Árvores binárias completas, 304–305
  altura de, 306
  número de vértices de, 306
Árvores binárias estendidas, 302
Árvores binomiais, 745
Árvores com raízes, 302, 685–688
  altura de, 691, 743
  árvores de decisão, 698–700
  balanceada, 691, 743
  B-árvore de grau *k*, 745
  binomial, 745
  definição de, 743
  ordem de nível de vértices de, 745
  ordenada, 687, 743, 745
  S_k-árvore , 745
Árvores de busca binária, 695–698, 743
Árvores de decisão, 698–700, 743
Árvores de derivação, 790–792, 839
Árvores de família, 683, 684
Árvores de Fibonacci enraizadas, 695
Árvores de jogos, 704–707, 743
Árvores geradoras, 724–734, 744
  construção
    por busca em largura, 729–731
    por busca em profundidade, 726–729
  de grau restrito, 746
  definição de, 724
  distância entre, 737
  distribuição múltipla em IP, 726
  enraizadas, 737
  máximas, 742
  mínimas, 738–741, 744
Árvores geradoras enraizadas, 737
Árvores geradoras mínimas, 737–741, 744
Árvores graciosas, 746
Árvores *quad*, 748
Asserção
  final, 323, 329
  inicial, 323, 329
Asserção final, 323, 329
Asserção inicial, 323, 329
AT&T, 625
Atletas de corrida de bicicleta, 424
ATM células, 144
Atores, 593
Auto-dual, 782
Autômato de pilha, 825
Autômato finito determinístico, 807–811, 839
Autômato finito equivalente, 806, 809–811
Autômatos finitos, 805–811
  conjunto não reconhecido por, 824–825
  conjuntos regulares reconhecidos por, 819–821
  determinísticos, 807–811, 839
  equivalentes, 809–811
  não determinísticos, 811–814, 820–821, 822, 839
Autômatos finitos não determinísticos, 811–814, 820, 822, 839
Autômatos linearmente limitados, 825
Autômatos
  de pilha, 825
  finitos, 805–811
    conjuntos regulares reconhecidos por, 819–821
    determinísticos, 807–811, 839
    não determinísticos, 811–814, 820–821, 822, 839
    um conjunto que não é reconhecido por, 824–825
  linearmente limitados, 825
  quociente, 817
Autoria conjunta, 593–594
Axioma(s), 75, 845–849
  completude, 846
  corpo, 845–846
  da indução matemática, 849
  na demonstração de fatos básicos, 846–849
  ordem, 846
  para números reais, 845–846
  para o conjunto dos inteiros positivos, 849
Axiomas de corpo, 845–846
Babbage, Charles, 27
Bachmann, Paul Gustav Heinrich, 182
Backtracking, 726–729, 744
  aplicações de, 731–732
  em grafos direcionais, 733
Backus, John, 792
Bacon, Kevin, 624
Bactéria, 457
Banco de dados
  chave composta de, 532
  chave primária de, 531
  intensão de, 531
  modelo relacional, 531–532, 581
  registro em, 531
Barco, 632
Barra de chocolate, 292
B-árvore de grau *k*, 745
Base *b*
  pseudoprimo forte para, 245
  Teste de Miller para, 245
Base de vértices, 632
BASIC, 340
Basquete, 382
Bayes, Thomas, 418, 420
Bell, E. T., 566
Bernoulli, James, 406, 407
Bicondicional, 9–10
  equivalência lógica para, 25
Bigit, 16
Binit, 16
Binômio de Newton, 363–365, 387
  estendido, 486
Biscoito, 373
Biscoitos da sorte, 388
Bit, 104
  origem do termo, 15, 16
Blocos de comandos, 855
BNF (Forma de Backus–Naur), 792–793, 839
Bomba de Arquimedes, 848
Boole, George, 3, 5, 420, 749
Boruvka, Otakar, 740
Bots, 734
Bottom-up parsing, 791
Bubble sort, 173, 257
  bidirecional, 173, 259
  complexidade de pior caso de, 194–195
  variante de, 173, 219
Bubble sort bidirecional, 173
Bug, computador, 811
Busca em largura, 729–731, 744
Busca em profundidade, 726–729, 744
  aplicações com, 732–734
  em grafos orientados, 732–734
Busca seqüencial, 170
Buscadores booleanos, 13
Butano, 688
Bytes, 220
Cabra, 632
Cachorros, 17
Cactus, 746
CAD, programas 772
Cadeia, 585
Cadeia de cartas, 691
Cadeia ternária, 458
Cadeia vazia, 151, 787
Cadeias, 151
  bit, 15, 104, 129–130
  comprimento ou extensão de, 15, 151, 454–455
    definição recursiva de, 301
  concatenação de, 300, 804, 839
  contagem, 336
    sem 0s consecutivos, 454–455
  de parênteses, balanceados, 332
  decodificação, 700–701
  definida recursivamente, 300
  distinguíveis, 827
  gerar a próxima maior, 385
  indistinguíveis, 827
  ordem lexicográfica de, 568
  reconhecida (aceita), 806, 836
  ternária, 458
  vazia, 151, 787
Cadeias balanceadas de parênteses, 332
Caixa de dinheiro, 372–373
Caixas, distribuição de objetos, 376–379
  distintos, 376
  idênticos, 376
Caixas de e-mail, 421
Cálculo predicado, 33
  equivalência lógica em, 39
Cálculo
  predicado, 31
  proposicional, 2
Caminhos, 622–624
  comprimento de, em um grafo com pesos, 648
  contando, entre vértices, 628–629
  de comprimento *n*, 623
  e isomorfismo de grafo, 627–628

em grafos de colaboração, 623
em grafos de relacionamento, 623
em grafos não orientados, 622
em grafos orientados, 546–547, 582, 676
em grafos simples, 622
em multigrafos orientados, 623
em um grafo de Hollywood, 623–624
eulerianos, 633–638, 676
hamiltonianos, 638–643, 676
mais curto, 647–655, 676
terminologia de, 622
Caminhos eulerianos, 633–638, 676
Caminhos hamiltonianos, 638–643, 676
Camisetas, 344
Campeonatos da Copa do Mundo de Futebol, 362
Campos, 531
Cântaro, 103
Cantor, Georg, 111, 113, 160
Caractere alfanumérico, 340
Cardinalidade
alef zero, 158
de conjunto, 113, 116, 158–160
Carga indutiva, 282, 293, 331
Carmichael, Robert Daniel, 240
Carregando a pergunta, 84, 105
Carroll, Lewis (Charles Dodgson), 44–45, 50
Casos, demonstrações por, 86–90, 105
Cavaleiros e bandidos, 13–14
Cavalos de raça, 27
Cayley, Arthur, 683, 688
Células adjacentes, 768
Centopéia, 746
Centro, 695
César, Júlio, 207
Chamadas telefônicas, 594
Chave
composta, 532
para hashing, 205
primária, 531–532
Chave composta, 532, 582
Chave primária, 531–532, 581
Chebyshev, Pafnuty Lvovich, 438
Chen, J. R., 215
Chi (letra grega), 667
Filho do vértice esquerdo, 687
Chomp, 91, 292, 586
Chomsky, Avram Noam, 785, 789, 791
Chung-Wu Ho, 290
Church, Alonzo, 833
Ciclo
com n vértices, 601
em grafos orientados, 546, 582
Ciclos eulerianos, 633–638, 676
Ciclos hamiltonianos, 636–643, 676
Ciência da computação, 792
Cilindro, 769
Cimitarras de Mohammed, 637
Circuito de saída múltipla, 764
Circuito simples, 627
Circuitos combinatórios, 761–763
exemplos de, 763–764
minimização de, 766–779
profundidade de, 766
Circuitos de chaves *hazard-free*, 784
Circuitos ou ciclos
combinatórios, 761–763
exemplos de, 763–764
minimização de, 766–779
profundidade de, 766
em grafo, 622
em grafo orientado, 546, 582
eulerianos, 633–638, 676
hamiltonianos, 638–643, 676
saída múltipla, 764
simples, 622
Classe
congruência, 204
equivalência, 558–559, 582
definição de, 558
e partições, 559–562
representante de, 558
NP, 197
P, 197
Classe de congruência, 204
Classes de equivalência, 558–559, 582
definição de, 558
e partições, 559–562
representante de, 558
Classificação de gramáticas de Chomsky, 789–790
Cláusula, 68
Clique, 678
COBOL, 792, 811
Codificação, 207, 231, 257
RSA, 242–243
Codificação binária decimal, 231
Codificação de Huffman, 701–703, 744
variações de, 703
Código de César, 207–208
Código de número de telefone, 337
Código de substituição, 208
Código
afim, 208
César, 207–208
RSA, 241–244
substituição, 208
Códigos de Gray, 642–643
Códigos de prefixo, 700–703, 743
Coeficiente multinomial, 382
Coeficiente(s)
binomial, 357, 363–368, 387
estendido, 486
constante
relações de recorrência linear e heterogênea com, 467–471, 514
relações de recorrência linear e homogênea com, 460, 461–467, 513-514
multinomial, 382
Coeficientes binomiais, 357, 363–368, 386
estendidos, 486
Coeficientes binomiais estendidos, 486
Coeficientes constantes
relações de recorrência lineares e heterogêneas com, 467–471, 514
relações de recorrência lineares e homogêneas com, 460, 461–467, 513
Coelhos, 451–452
Colisão, 206
em funções de hashing, probabilidade de, 410–411
Coloração
de grafos, 666–672, 676
de mapas, 666–667
em grafos bipartidos, 603
em número cromático, 667
Coloração *k*-upla, 674
Coluna de matriz, 247
Comando(s), 854
"se e somente se," 8, 9
begin, 855, 858
blocos de, 855
condicionais, 6
para correção de programa, 324–326
de atribuição de valores, 854–855
end, 855, 858
procedure, 855
regras de inferência para, 71–72
se, então, 6, 8
Comando **begin**, 855, 858
Comandos de atribuição de valores, 854–855
Comandos de procedimento, 854, 858–859
Comandos **end**, 855, 858
Combinação linear, como mdc, 232
Combinações, 357–360
com repetição, 371–375
de eventos, 396–398, 403–404
gerando, 384–385
linear, 232
Combinações generalizadas, 371–375
Combinatória, 335, 352, 378, 386
Comentários em pseudocódigo, 855–856
Compiladores, 559, 810
Complemento, 749, 781
da função booleana, 751
de conjunto, 123, 163
de conjunto fuzzy, 132-133
de dois, 230–231
de um, 230
duplo, propriedade do, em álgebra booleana, 753, 755, 757
Completude funcional, 759
Complexidade computacional de algoritmos, 193
de algoritmo de Dijkstra, 653
máquina de Turing para tornar precisa, 833-834
Complexidade da merge sort, 479
Complexidade de algoritmos constante, 197
Complexidade de caso médio de algoritmos, 195, 431–433
Complexidade de pior caso de algoritmos, 194, 195–196, 197
Complexidade de tempo de algoritmos, 193–196
Complexidade dos algoritmos, 193–199
caso médio, 195, 431–433
computacional, 193
de algoritmo de Dijkstra, 653
constante, 197
da merge sort, 479
de algoritmos de busca, 194–195, 197
de algoritmos de busca binária, 194–195
de algoritmos de busca em largura, 731

de algoritmos de busca em profundidade, 729
de algoritmos de busca linear, 194
de algoritmos de ordenação, 195–196
de bubble sort, 194–195
de insertion sort, 196
de maior elemento encontrado em um conjunto, 194
espacial, 193
exponencial, 197
fatorial, 197
linear, 197
logarítmica, 197
pior caso, 194, 195–196
polinomial, 197
temporal, 193–196
Complexidade espacial de algoritmos, 193
Complexidade exponencial de algoritmos, 197
Complexidade fatorial de algoritmos, 197
Complexidade linear de algoritmos, 197
Complexidade logarítmica de algoritmos, 197
Complexidade polinomial de algoritmos, 197
Componentes conexas de grafos, 625–626
fortemente, 627, 676
Componentes fortemente conexas de um grafo, 627, 676
Componentes fortemente conexas gigantes (GSCC), 627
Composição de funções, 139–141, 163
Composta de relações, 526–527, 581
Compressão de dados, 701
Computação, modelos de, 785–837
Concatenação, 300, 804, 839
Conclusão, 6, 44, 64
falácia da afirmação, 69
Condições
don't care, 774–775
iniciais, 450, 513
Condições *don't care*, 774–775
Condições iniciais, 450, 513
Conectivo de Sheffer, 28, 29
Conectivos, 4
Conectivos lógicos, 4
Congruência módulo *m*, 203–204, 556, 558, 582
inverso de, 234
Congruências, 205–207
lineares, 234–235, 257
Congruências lineares, 234–235, 257
Conjectura, 75, 105
de Frame, 460
de Goldbach, 214–215
e contra-exemplos, 96–97
e demonstração, 96–97
números primos gêmeos, 215
$3x + 1$, 101–102
Conjectura de Frame, 460
Conjectura de Goldbach, 214–215
Conjectura dos números primos gêmeos, 215
Conjectura $3x + 1$, 101–102
Conjunção, 4–5, 104, 781
lei distributiva da disjunção sobre, 23
Conjunção, reticulado, 585
Conjunto(s), 111–121
bem ordenado, 568, 582
cardinalidade de, 116, 158–160, 163
complemento de, 123, 163
conjunto de guarda, 675
contável, 158, 218
corte, 746
de partes, 116–117, 163
definição de, 111, 163
definido recursivamente, 299–307, 329
demonstrações de fatos sobre, por indução matemática, 273
descrição de, 111–112
diferença de, 123, 163
simétrica, 131, 163
diferença simétrica de, 131, 163
disjunto, 122
dominante, 679
elementos de, 112, 163
finito, 116, 158, 163, 500, 514
combinações de, 384–385
subconjuntos
contagem de, 338
número de, 273–274
união de dois, número de elementos em, 500, 514
união de três, número de elementos em, 501–503, 514
fuzzy, 132–133
complemento de, 133
interseção de, 133
união de, 133
identidades de, 124–126
igual, 113, 163
imagem de, 136
imagem inversa de, 147
incontável, 160
infinito, 116, 159, 163
interseção de, 121–122, 126–128, 163
linearmente ordenado, 568, 582
membros de, 112, 163
não reconhecido por um autômato finito, 824–825
operações com, 121–130
parcialmente ordenado, 566, 582
bem fundado, 581
denso, 581
digrama de Hasse de, 571–572, 573, 582
dual de, 579
elemento maximal de, 572, 582
elemento minimal de, 572, 573, 582
elementos comparáveis em, 567, 582
elementos incomparáveis de, 567, 568, 582
em anticadeia, 585
em cadeia, 585
limitante superior de, 574, 582
maior elemento de, 573, 582
menor elemento de, 573, 582
menor limitante de, 574, 582
partição de, 560–561, 582
reconhecido pelas máquinas Turing, 830–831
reconhecido pelo autômato de conjunto finito, 819–821
regular, 817–819, 820, 821–824, 839
relação em, 118, 521–522
representação computacional de, 129–130
sucessor de, 132
tipo, 113
totalmente ordenado, 568, 582
união de, 121, 126–128, 163
unitário, 114
universo, 113, 163
vazio (nulo), 114, 163
verdade, 119
Conjunto bem ordenado, 332, 568, 582
Conjunto de corte de grafo, 746
Conjunto de guarda, 675
Conjunto de partes, 116–117, 163
Conjunto disjunto, 122
Conjunto dominante, 679
Conjunto dominante mínimo, 679
Conjunto enumerável, 158
Conjunto funcionalmente completo de operadores
para funções booleanas, 759, 781
para proposições, 29
Conjunto fuzzy, 132–133
complemento de, 133
intersecção de, 133
união de, 133
Conjunto imagem, 134, 163
Conjunto incontável, 160
Conjunto independente de vértices, 680
Conjunto infinito, 116, 158, 163
Conjunto linearmente ordenado, 568, 582
Conjunto nulo, 114, 163
Conjunto parcialmente ordenado, 566, 582
anticadeia, em, 585
bem fundado, 581
cadeia, em, 585
denso, 581
diagrama Hasse de, 571–572, 573, 582
dual de, 579
elemento maximal de, 572, 582
elemento minimal de, 572, 573, 582
elementos comparáveis em, 568, 582
elementos incomparáveis em, 568, 582
maior elemento de, 573, 583
maior limite de, 574, 582
menor elemento de, 573, 582
menor limite de, 574, 582
Conjunto totalmente ordenado, 568, 582
Conjunto unitário, 114
Conjunto universo, 113, 163
Conjunto vazio, 114, 163
Conjuntos finitos, 116, 158, 163, 500, 514
subconjuntos de
contagem, 338
número de, 273–274
união de dois, número de elementos em, 500, 514
união de três, número de elementos em, 501–503, 514
Conjuntos regulares, 817–819, 820, 839
Conjunto-verdade, 119
Conseqüência, 6
Consistência, de sistemas de especificações, 12

Construção de números reais, 849
Construção do conjunto, 112
Construções
  condicional, 856–857
  laço, 857–858
Construções condicionais, 856–857
Construções de laço, 857–858
Contagem, 335–391, 449–518
  caminhos entre vértices, 628–629
  com repetição, 370–371
  combinações, 357–360
  diagrama de árvores para, 343–344
  endereços de Internet, 341
  funções, 336–337
  funções booleanas, 751
  funções generalizadas para, 488–493
  funções sobrejetivas, 337
  funções sobrejetoras, 509–510, 514
  nomes de variáveis, 340
  números de telefone, 337
  permutações, 355–357
  permutações caóticas, 510, 514
  por distribuição de objetos em caixas, 376–379
  princípio da casa dos pombos e, 347–353
  princípios básicos de, 335–340
  relações, 522
  relações reflexivas, 524
  senhas, 340
  seqüências de, 336
    sem 0s consecutivos, 454–455
  subconjuntos de um conjunto finito, 338
Contagem, de números racionais, 158, 163, 218
Contingência, 21, 104
Contradição, 21, 80
  demonstração por, 80–83
Contradomínio de função, 134, 149, 163
Contra-exemplos, 34, 83
  conjectura e, 96–97
Contrapositiva, de implicação, 8, 104
Controle de qualidade, 412
Conversão de base, 220
Cor margeada, 674
Corolário, 75, 105
Correspondência um para um (bijetora), 138, 159, 163
Covariância, 441
CPM (Critical Path Method), 587
Crawlers, 734
Crescimento de funções, 180–190
Criptologia, 207–208
  chave privada, 241
  chave pública, 231, 241–244
Criptosistema de chave pública, 231, 241–244, 257
Criptossistemas de chave privada, 241
Críquete, 93
Critical Path Method (CPM), 587
Crivo de Eratóstenes, 210, 507–509, 514
Curso de Matemática Pura, Um (Hardy), 93
Czekanowski, Jan, 740
Dado, 416, 436–437
  dodecaedro, 443
  octaedro, 443
Dados, 394
Damas, árvore de jogo para, 707
Datagrama de internet, 346–347
Datagramas, 346–347
Datalogia, 792
de Méré, Chevalier, 400
De Morgan, Augustus, 22, 25, 668, 669
Decimal, codificação binária, 231
Décimo problema de Hilbert, 835
Decodificação, 208, 231, 257
  RSA, 243
Decomposição funcional, 784
Deep Blue, 707, 748
Definição abstrata de álgebra booleana, 755
Definição de probabilidade de Laplace, 394
Definição recursiva, 206
Definição
  domínio de, 149
  recursiva, 263, 294–308
Definições indutivas, 295–308
  de cadeias, 300
  de conjuntos, 299–307, 328
  de estruturas, 299–307
  de fatoriais, 296
  de funções, 295–299, 328
Definições recursivas, 263, 294–308
  de árvores binárias completas, 303
  de árvores binárias estendidas, 302
  de conjuntos, 299–307, 328
  de estruturas, 299–307
  de fatoriais, 311–312
  de funções, 295–299, 328
  de seqüência, 300
Demonstração(ões)
  combinatorial, 359, 386
  conjectura e, 96–97
  de algoritmos recursivos, 315–317
  de equivalência, 82–83
  de existência, adaptando, 96
  definição de, 75, 105
  direta, 76–77, 105
  erros em, 83–84
  existência, 91–92, 105
  indireta, 77
  métodos de, 75–102, 105
  por casos, 86–90, 105
  por contradição, 80–83, 105
  por exaustão, 87
  por indução estrutural, 304
  por indução matemática, 266–279
  por trivialização, 78, 79, 105
  por vacuidade, 78, 105
  unicidade, 92–93, 105
Demonstração combinatória, 359, 386
Demonstração de existência não construtiva, 91, 105
Demonstração de teoremas automatizada, 108
Demonstração direta, 76–77, 105
Demonstração indireta, 77
Demonstração por exaustão, 87
Demonstração por trivialização, 78, 79, 105
Demonstração por vacuidade, 78, 105
Demonstrações de existência, 91–92, 105
  construtiva, 91
  não construtiva, 91
Demonstrações de existência construtiva, 91, 105
Demonstrações de unicidade, 92–93, 105
Denso
  grafo, 614
  poset, 581
Derivação, 787
Desafio(s)
  "Uma Viagem ao Redor do Mundo", 639
  da travessia do rio, 632
  da zebra, 21
  das crianças enlameadas, 14–15
  das três portas de Monty Hall, 398, 425
  de Reve, 454
  Icosian, 639, 640
  lógico, 13–15
  paradoxo do aniversário, 409–411
  quebra-cabeça, 292
  Torre de Hanói, 452–454
Desafio de Hamilton "Uma Viagem ao Redor do Mundo", 639
Descartes, René, 117
Descendentes de vértice, 686, 743
Designação de freqüência, 672
Desigualdade
  de Chebyshev, 438–439, 442
  de Markov, 441
  triangular, 102
Destacamento, propriedade de, 65
Desvio-padrão, 436
Diagnóstico de resultados de testes, 419–420
Diagonal de um polígono, 288
Diagrama de árvore, 343–344, 386
Diagrama de estados para máquina de estado finito, 798
Diagramas
  árvore, 343–344, 386
  estado, para máquina de estado finito, 798
  Hasse, 571–572, 574, 575, 582
  Venn, 113, 114, 115, 122, 123, 163
Diâmetro de grafo, 680
Diferença, 850
  atrasada, 460
  comum, 151
  de conjuntos, 123, 163
    simétrica, 131, 163
  de multiconjuntos, 132
  progressiva, 516
Diferença simétrica, de conjuntos, 131, 163
Diferenças atrasadas, 460
Diferenças progressivas, 516
Dígito binário
  origem do termo, 15
Dígitos
  binário
    origem do termo, 15, 16
  Cantor, 386
Dígitos de Cantor, 386
Dígrafos, 582, 591, 675
  arestas de, 591, 600–601
  auto-inverso, 679
  busca em profundidade em, 732–734
  caminhos em, 546–547, 582, 676
  ciclo de Euler de, 634
  circuito (ciclo) em, 546, 582

conexidade em, 626–627
laços em, 541, 582
representação de relações usando, 541–542
reverso de, 611
vértice de, 590, 600
Dijkstra, Edsger Wybe, 649
Diofanto, 101, 239
Dirac, G. A., 641
Dirichlet, G. Lejeune, 348
Discours (Descartes), 117
Discurso, universo de, 34
Disjunção, 4–5, 104, 781
propriedade associativa de, 24
propriedade distributiva da, sobre conjunção, 23
Distância
entre árvores geradoras, 737
entre vértices distintos, 680
Distintos
caixas, 376
objetos, 376
Distribuição
binomial, 406–408
de probabilidade, 401
de variável aleatória, 408, 442
geométrica, 433–434, 442
uniforme, 402, 442
Distribuição binomial, 406–408
Distribuição de objetos em caixas, 376–379
Distribuição de probabilidade, 401
Distribuição de trabalhos, 510
Distribuição geométrica, 433–434, 442
Distribuição múltipla em IP, 726
Distribuição única, 726
Distribuição uniforme, 402, 442
div, 202, 225–226
Dividendo, 214
Divine Benevolence (Bayes), 420
Divisão
de números inteiros, 200–208
tentativa, 214
Divisão e resolução
algoritmos, 474, 514
relações de recorrência, 474–482
Divisor, 201, 214
máximo comum, 215–217, 232–234, 258
Divisor comum, máximo, 215–217
Dodgson, Charles (Lewis Carroll), 44–45
"Dodecaedro do Viajante," 639, 640
Domínio
de definição, 149
de discurso, 34
de relação, 530
de relação n-ária, 530
de um quantificador, 34
de uma função, 134, 149, 163
universo de, 34
Dominós, 97, 266
Dual
de expressão booleana, 754
de função booleana, 754
de poset, 579
de proposição composta, 29
Dualidade em reticulado, 587
Dudeney, Henry, 454
Duplo complemento, propriedade do, em álgebra booleana, 753, 757
E (AND), 13, 15–16
EBNF (Forma de Backus–Naur estendida), 796
Ecologia, grafo de superposição de nichos em, 592
Einstein, Albert, 21
Elemento(s), 112, 163
comparáveis, em conjunto parcialmente ordenado, 567, 582
de matriz, 247
equivalentes, 556
fixos, 441
incomparáveis, em conjunto parcialmente ordenado, 567, 582
maior, de conjunto parcialmente ordenado, 573, 582
maximal, de conjunto parcialmente ordenado, 572, 582
menor, de conjunto parcialmente ordenado, 573, 582
minimal, de conjunto parcialmente ordenado, 572, 573, 582
Elemento maximal de um poset, 572, 582
Elemento minimal de um poset, 572, 582
Elementos, 112, 163
Elementos, Os (Euclides), 212
Elementos comparáveis em poset, 567, 582
Elementos equivalentes, 556
Elementos fixos, 441
Elementos incomparáveis em poset, 567, 582
Elements of Mathematical Logic (Lukasiewicz), 721
Emparelhamento maximal, 605
Empréstimo, 457
Encontro, em reticulados, 585
Encontro de matrizes zero-um, 252
Endereços de Classe A, 341
Endereços de Classe B, 341
Endereços de Classe C, 341
Endereços de Classe D, 341
Endereços de Classe E, 341
Endereços de internet, contagem, 341
Engenho analítico, 27
Enigma da zebra, 21
Entrada de matriz, 247
Enumeração, 335, 386
códigos, 455–456
Enumeração de códigos, 455–456
Equação
característica, 461
de diferença, 460
Equação característica, 461
Equação de diferença, 460
Equações lineares simultâneas, 256
Equivalência, demonstrações de, 82–83
Equivalências lógicas, 22–27, 39, 105
em cálculo predicado, 39
em cálculo proposicional, 22–27
Equivalências proposicionais, 21–30
Eratóstenes, 507–508
crivo de, 210, 514
Erdös, Paul, 584, 623
Erros, em demonstrações, 83–84
por indução matemática, 278–279
Erros comuns
em demonstrações exaustivas, 90
em demonstrações por casos, 90
Erros de programação, 811
Escada infinita, 263–264
por indução completa, 284
Escopo de um quantificador, 38
Espaço amostral, 394
Especificações, sistema de, 12, 43–44
Espessura de um grafo, 666
Estado atingível, 841
Estado inicial, 798, 805
Estado transiente, 841
Estados, 798
alcançável, 841
transiente, 841
Esteróides, 424
Estratégia de demonstração, 79, 86–102
Estratégia minimax, 706, 743
Estruturas, definidas recursivamente, 299–307
Euclides, 212, 227, 228
Euler, Leonhard, 214, 633, 635
"Eureca", 848
Evento(s), 394
combinações de, 396–398, 403–404
complementar, 396
independente, 399, 405–406, 442
mutuamente independente, 444
Evento complementar, 396
Eventos independentes, 399, 400, 405–406, 442
Eventos mutuamente independentes, 444
Exames, planejamento de horário, 671, 672
Exames finais, planejamento de horário, 671, 672
Exatidão de programas, 322–328, 329
laços invariáveis, 326–327
parcial, 323
proposições condicionais para, 324–326
regras de inferência para, 323–324
verificação de programa para, 323
Exatidão parcial, 323
Exatidão
de algoritmos recursivos, 315–316
de programas, 322–328, 329
laços invariáveis, 326–327
parcial, 323
proposições condicionais para, 324–326
regras de inferência, 323–324
verificação, 323
Excentricidade do vértice, 695
Expansão binária, 219
Expansão complemento de dois, 230–231
Expansão complemento de um, 230
Expansão de Cantor, 231, 386
Expansão decimal codificada binária, 774
Expansão em produto de somas, 758
Expansão hexadecimal, 219–220
Expansão na base *b*, 219, 221
Expansão octal, 220, 257
Expansão ternária balanceada, 230
Expansão

binária, 219, 257
de Cantor, 231
de dois complementos, 230–231
de um complemento, 230
decimal codificada binária, 774
hexadecimal, 219–220, 257
na base *b*, 219, 221
octal, 220, 257
ternária balanceada, 230
Expansões em soma de produtos, 758–759, 781
simplificação, 769–779
Expansões ternárias balanceadas, 230
Experimento, 394
Exponenciação modular recursiva, 313
Expressão(ões)
binomial, 363
booleana, 750–752, 781
forma infixa de, 719
forma pós-fixa de, 721
forma prefixa de, 719
lógica
traduzindo para sentenças do português, 55
traduzindo sentenças em português para, 11, 42–43, 56–57
regular, 818, 839
Expressão binomial, 363
Expressões booleanas, 750–752, 781
dual de, 754, 781
Expressões booleanas equivalentes, 750, 751
Expressões lógicas
traduzindo para sentenças do português, 42–43
traduzindo sentenças do português para, 11,42–43, 56–57
Expressões regulares, 818, 839
Extensão de cadeia, 151
definição recursiva de, 301
relação "menor ou igual a", 567
Extensão de cadeia de bit, 15, 454–455
Extensão de caminho
em grafo com pesos, 648
em grafo orientado, 546
Exterior de polígono simples, 288
Extremidades da aresta, 598
Falácia, 63, 69, 105
carregando a pergunta, 84, 105
da afirmação da conclusão, 69
da negação das hipóteses, 69
de raciocínio circular, 84, 105
Falso
negativo, 419, 421
positivo, 419, 421
Fator, de números inteiros, 201
Fatoração de números primos, 211–212
para encontrar o máximo divisor comum, 216
para encontrar o mínimo múltiplo comum, 216–217
unicidade de, 233–234
Fatoriais, procedimento recursivo para, 311–312
Fatos, 75
Fatos de divisibilidade, demonstração, por indução matemática, 273
Fatos Prolog, 45
Fazendeiro, 632
Fecho de Kleene, 804–805, 817, 839
Fecho de relações, 544–553, 582
reflexivo, 544, 582
simétrico, 545, 582
transitivo, 544, 547–550, 582
cálculo, 550–553
Fecho reflexivo de relação, 544, 582
Fecho simétrico de relação, 545, 582
Fecho transitivo da relação, 544, 547–550, 582
cálculo, 550–553
Fermat, Pierre de, 239
Fibonacci, 298
Filho direito, 687
Filho do vértice, 686, 743
Filtro, spam, 421–423
Filtros de spam de Bayes, 421–423
Flavio Josefo, 459
Flecha de Peirce, 29
Flip-flops, 784
Florestas, 684, 685
definição de, 742
geradora, 736
geradora mínima, 742
Flutuação, princípio de, 848
Fluxo de informação, modelo reticulado do, 575–576
FNP (forma normal prenex), 62
Folha, 686, 743
Forma(s)
Backus–Naur, 792–793, 839
infixa, 719
normal conjuntiva, 758
normal disjuntiva
para proposições, 29
para variáveis booleanas, 758
normal prenex, 62
pós-fixa, 720
prefixa, 719
Forma Backus–Naur (BNF), 792–793, 839
Forma de Backus–Naur estendida (EBNF), 796
Forma derivável, 787
Forma diretamente derivável, 787
Forma infixa, 719
Forma normal conjuntiva, 758
Forma normal disjuntiva
para proposições, 29
para variáveis Booleanas, 758
Forma normal prenex (FNP) , 62
Forma pós-fixa, 720
Forma prefixa, 719
Fórmula(s)
bem formada, 301
de operações, 301
para proposições compostas, 301
de Euler, 659–663, 676
de somatório, 157, 267
de Stirling, 146
para números de Fibonacci, 463–464
Fórmula bem formada, 301
de operadores, 301
em notação prefixa, 724
indução estrutural e, 305
para proposições compostas, 301
Fórmula bem formada, 724
Fórmula de Euler, 659–663, 676
Fórmula de Stirling, 146
Fórmulas de somatório
demonstração por indução matemática, 267–270
FORTRAN, 792
Fração, unidade (egípcia), 331
Fração egípcia (unidade), 331
Fracasso, 406
Frend
Sophia, 25
William, 27
Frisbee movido a foguete, 750
Função(ões), 133–146, 163
91 de McCarthy, 331
adição de, 135–136
assintótica, 192
bijetora, 138, 163
booleana, 749–755, 781
auto-dual, 782
conjunto funcionalmente completo de operadores para, 759
dual de, 754
implicante de, 770
minimização de, 766–779, 781
representação, 757–759
threshold, 783
busy beaver, 835
calculada por máquinas Turing, 831–832
característica, 148
como relações, 520
completa, 149
composições de, 139–141, 163
computável, 163, 835
contagem, 336–337
contradomínio de, 134, 149, 163
crescente, 137
crescimento de, 180–190
crescimento de combinações de, 186–189
de Ackermann, 310
de avaliação, 707
decrescente, 137
definição de, 133, 163
domínio de, 134, 149, 163
estritamente crescente, 137
estritamente decrescente, 137
exponencial, 851
fatorial, 145
generalizada, 484–495, 514
de probabilidade, 499
exponencial, 498
para contagem, 488–493
para demonstrar identidades, 495
para relações de recorrência, 493–495
gráficos de, 142
hashing, 205–206
colisão em, probabilidade de, 410–411
identidade, 138
imagem da, 134, 163
indefinida, 149
inversa, 139–140, 163

inversível, 140
iterada, 311
logarítmica, 851–853
logaritmo iterado, 311
maior inteiro menor que, 143–145, 163
maior número inteiro, 143–145
menor inteiro maior que, 143–145, 163
multiplicação de, 135–136
number-theoretic, 831
parcial, 149
produto de, 135–136
proposicional, 30, 31, 105
quantificação existencial de, 36
quantificação universal de, 33
recursiva, 133, 295–299, 328
sobrejetora, 137–138, 163
número de, 509–510, 514
soma de, 135–136
threshold, 783
transição, 798
um para um (injetora), 136, 138, 163
contagem, 337
ϕ de Euler, 218
Função bijetora, 138, 163
Função *busy beaver*, 835
Função característica, 148
Função chão, 143–145, 163
Função crescente, 137
Função de Ackermann, 310
Função de avaliação, 707
Função de transição, 798
extensão, 805
Função estritamente crescente, 137
Função estritamente decrescente, 137
Função exponencial generalizada, 498
Função fatorial, 145
Função generalizada, 484–495, 514
de probabilidade, 499
exponencial, 498
para contagem, 488–493
para demonstrar identidades, 495
para relações de recorrência, 493–495
Função generalizada ordinária, 484
Função identidade, 138
Função injetora (um a um), 136, 138, 139, 163
contagem, 337
Função inversível, 140
Função iterada, 311
Função logarítmica, 851–853
Função 91 de McCarthy, 331
Função maior inteiro menor que, 143–145, 163
Função parcial, 149
Função proposicional, 31, 32, 105
quantificação existencial de, 36
quantificação universal de, 34
Função sobrejetora, 137–138, 139, 163
número de, 509–510, 514
Função teto, 143–145, 163
Função *threshold*, 783
Função total, 149
Função um para um (injetora), 136, 139, 163
contagem, 337
Função ϕ de Euler, 218
Funções assintóticas, 146, 192
Funções booleanas, 749–755, 781
auto-dual, 782
dual de, 754, 781
implicante de, 770
minimização de, 766–779, 781
para o conjunto de operadores funcionalmente completo , 759
representação, 757–759
threshold, 783
Funções computáveis, 163, 835
Funções de hashing, 205–206
colisão em, probabilidade de, 410–411
Funções de number-theoretic, 831
Funções decrescentes, 137
Funções exponenciais, 851
Funções indefinidas, 149
Funções sobrejetoras, 137–138, 139, 163
número de, 509–510, 514
Fundir duas listas, 319
Galeria de Arte
problema, 675
Teorema, 675
Galinhas, 680
Gauss, Karl Friedrich, 204, 213
Generalização
existencial, 70, 71
universal, 70, 71
Geometria computacional, 288–290, 479–481
Gerador multiplicativo puro, 207
GIMPS (Great Internet Mersenne Prime Search), 213
Gödel, Escher, Bach (Hofstader), 333
Goldbach, Christian, 214, 215
Golfe, primeiro buraco, 279
Golomb, Solomon, 99
Google, 13, 734
Grafo(s), 589–682
auto-complementar, 620
bipartido, 602–604, 675
bipartido completo, 604, 675
caminhos em, 622–624
coloração, 666–672, 676
com peso, 676
caminho mais curto entre, 647–653
problema do caixeiro-viajante com, 653–655
complementar, 611
completo, 601, 675
componentes conexas de, 625–626, 676
conectividade em, 621–629
conjunto de corte de, 746
cromaticamente k-crítico, 674
da Web, 594–595
componentes fortemente conexas de, 627
de chamada, 594
componentes conexas de, 625
de colaboração, 593–594
de funções, 142
de influência, 593, 594
de interseção, 596
de precedência, 595
de relacionamento, 592–593
de superposição de nichos, 592, 593
definição de, 591
diâmetro de, 680
dual, 667
em modelo de mapas rodoviários, 595
emparelhamento em, 605
escala de integração muito grande, 682
esparso, 614, 741
finito, 589
Hollywood, 593
homeomorfos, 663–665, 676
infinito, 589
isomorfos, 611, 615–618, 676
caminhos e, 627–628
largura de banda de, 680
misto, 591
modelos, 592–595
m-partido completo, 678
multigrafo orientado, 592, 675
multigrafos, 590, 675
não orientado, 592, 599, 676
caminho euleriano de, 638
caminhos em, 622, 676
ciclo Euleriano de, 634
conexidade em, 624–626
orientação de, 679
subjacente, 601, 675
não planar, 663–665
n-regular, 610
número cromático de, 667
número de independência de, 680
número de livro de, 682
orientado, 582, 591, 675
aresta de, 591, 600–601
auto-inverso, 679
busca em profundidade em, 732–734
caminhos em, 546–547, 582, 676
ciclo euleriano de, 634
circuito (ciclo) em, 546, 582
conexidade em, 626–627
denso, 614
em laços, 541, 582
representação de relações usando, 541–542
reverso de, 611
simples, 591
vértice de, 590, 600
orientado fortemente conexo, 626
orientado simples, 591
orientável, 679
planar, 657–665, 676
propriedade monótona crescente de, 681
propriedade monótona decrescente de, 681
pseudografo, 590, 592, 675
raio de, 680
regular, 610, 676
representação, 611–618
rodas, 601, 676
simples, 590, 601–602, 675
aleatório, 680–681
aresta de, 590, 608
auto-complementar, 620
caminhos em, 725
coloração de, 667
com árvores geradoras, 725–726
denso, 614

esparso, 614
espessura de, 666
isomorfo, 615–618, 676
número de cruzamentos de, 666
orientação de, 679
planar conexo, 659–663
vértices de, 590, 608
subgrafo de, 607, 676
terminologia de, 597–601
união de, 661, 676
Grafo auto-complementar, 620
Grafo bipartido completo, 604, 675
Grafo complementar, 611
Grafo da Web, 594–595, 734
componentes fortemente conexas de, 627
Grafo da World Wide Web, 594–595
Grafo de chamadas telefônicas, 625
Grafo de Hollywood, 593
caminhos em, 623–624
Grafo de interseção, 596
Grafo de relacionamento, 592–593
caminhos em, 623
Grafo de superposição de nicho, 592, 593
Grafo dual, 667
Grafo finito, 589
Grafo infinito, 589
Grafo misto, 591
Grafo *m*-partido completo, 678
Grafo não orientado subjacente, 601, 675
Grafo *n*-regular, 610
Grafo orientado auto-inverso, 679
Grafo orientado simples, 591
Grafo regular, 610
Grafo simples aleatório, 680–681
Grafos bipartidos, 602–604, 675
coloração de, 603
Grafos com escalas muito grandes de integração (VLSI), 682
Grafos com peso, 676
árvore de extensão mínima para, 738–741
caminho mais curto entre, 647–653
problema do caixeiro-viajante com, 653–655
Grafos completos, 601, 675
Grafos conexos, 621–629
fortemente, 626
fracamente, 626
não orientados, 624–626
orientados, 626–627
simples planar, 659–663
Grafos de chamada, 594
componentes conexas de, 625
Grafos de colaboração, 593–594
caminhos em, 623
Grafos de influência, 593, 594
Grafos de precedência, 595
Grafos esparsos, 614, 741
Grafos fortemente conexos, 626
Grafos fracamente conexos, 626
Grafos homeomorfos, 663–665, 676
Grafos isomorfos, 611, 615–618, 676
caminhos e, 627–628
Grafos não orientados, 592, 599, 675
caminho euleriano de, 638
caminhos em, 622, 676
ciclo euleriano de, 634
conexidade em, 624–626
orientação de, 679
subjacentes, 601, 675
Grafos orientados, 582, 591, 675
arestas de, 591, 600–601
auto-inverso, 679
busca em profundidade em, 732–734
caminhos em, 546–547, 582, 676
ciclo de euleriano de, 634
circuito (ciclo) em, 546, 582
conexidade em, 626–627
laços em, 541, 582
representação de relações usando, 541–542
reverso de, 611
vértice de, 590, 600
Grafos orientáveis, 679
Grafos planares, 657–665, 676
Grafos que não são planares, 663–665
Grafos simples, 590, 601–604, 675
aleatório, 680–681
aresta de, 590, 608
auto-complementar, 620
caminhos em, 622
coloração de, 666
com árvores de extensão, 725–726
denso, 614
esparso, 614
espessura de, 666
isomorfos, 615–618, 676
número de cruzamentos de, 666
orientação de, 679
plano conectado, 659–663
vértices de, 590, 608
Graham, Ron, 584
Gramática ambígua, 840
Gramática de tipo 0, 789, 838
Gramática de tipo 1, 789, 838–839
Gramática de tipo 2, 789, 839
Gramática de tipo 3, 789, 821–824, 839
Gramática não contratora, 790
Gramática sensível ao contexto, 789
Gramática(s), 785–793
ambígua, 840
estrutura de frase, 787–790, 838
forma de Backus–Naur, 792–793, 839
livre de contexto (tipo 2), 789, 839
monótona, 790
não contratora, 790
produções de, 787, 838
regular (tipo 3), 789, 817, 821–824, 839
sensível ao contexto, 789
tipo 0, 789, 838
tipo 1, 789, 838
Gramáticas de estrutura de frase, 787–790, 838
Gramáticas livres de contexto, 789, 839
Gramáticas monótonas, 790
Gramáticas regulares, 789, 817, 821–824, 839
Grandes números, lei dos, 407
Grau
de região, 661
de relações de recorrência lineares e homogêneas, 461
de pertinência, 132
de relações n-árias, 530
de vértice de grafo não orientado, 598
Grau de entrada de vértice, 600, 675
Grau de saída de vértice, 600, 675
Gray, Frank, 643
Great Internet Mersenne Prime Search (GIMPS), 213
Gripe aviária, 424
GSCC (componente fortemente conexa gigante), 627
Guare, John, 623
Guerra Civil, 32
Guerra de tortas, ímpar, 276
Guerra entre judeus e romanos, 459
Guthrie, Francis, 669
Hadamard, Jacques, 213
Haken, Wolfgang, 668
Hamilton, William Rowan, 640, 669
Hanói, torre de, 452–454
Hardy, Godfrey Harold, 93
Hasse, Helmut, 571
Heaps, 748
Heawood, Percy, 668
Hérmias, 2
Hidrocarbonetos, modelo de árvores, 688
Hidrocarbonetos saturados, modelo de árvore, 688
Hipercubo, 607
Hipótese, 6
falácia de negação, 69
indutiva, 265
HIV, 424
Ho, Chung-Wu, 290
Hoare, C. Anthony R., 323, 324
Hofstader, Douglas, 333
Hopper, Grace Brewster Murray, 811
Hops, 606
HTML, 796
Huffman, David A., 701, 702
"Icosian Game", 639, 640
Ideal para os pretendentes, tarefa estável, 293
Idênticos
caixas, 376–379
objetos, 375, 376–379
objetos, com permutações, 375
Identidade
combinatorial, prova de, 359, 386
de Pascal, 366–367, 387
de Vandermonde, 367-368
hexágono, 369
Identidade de Pascal, 366–367, 387
Identidade de Vandermonde, 367–368
Identidade do hexágono, 369
Identidades, combinatorial, 364–368
Identidades combinatórias, 364–368
Identificador, 559, 793
Igualdade, de conjuntos, 113, 163
Ilha de Kneiphof, 633
Imagem, 136, 163
de conjunto, 136
inversa, de conjunto, 147
Imagem inversa do conjunto, 147
demonstração, funções generalizadas para, 495

Implicação oposta, 8, 104
Implicações, 6–10, 105
  equivalências lógicas para, 26–27
Implicante, 770
  número primo, 770
    essencial, 770
Implicante primo, 770, 781
  essencial, 770
Implicante primo essencial, 770, 781
Indicador de variável aleatória, 440
Índice, de somatória, 153
Indução
  completa, 283–285, 328
  da boa ordem, 568–569, 582
  demonstrações por, 266–278
  estrutural, 304–306, 328
  generalizada, 307–308
  incompleta, 284
  matemática, 25, 263–279
  princípio de, 328
  segundo princípio de, 284
  validade de, 278
Indução bem fundada, princípio da, 585
Indução completa, 283–285, 328
Indução da boa ordem, 582
  princípio de, 568–569
Indução estrutural, 303–306, 329
  demonstração por, 304
Indução generalizada, 307–308
Indução incompleta, 284
Indução matemática, 25, 263–279
  axioma de, 849
  carga indutiva com, 282
  demonstrações
    de fatos de divisibilidade, 273
    de fórmulas de somatório, 267
    de inequações, 271
    de resultados sobre algoritmos, 275
    de resultados sobre conjuntos, 273–275
    erros em, 278–279
    por, 266–278
  estrutural, 303–306, 328
  segundo princípio de, 284
  validade de, 278
  erros em, 278–279
  generalizada, 307–308
  princípio de, 328
Inequação
  de Bernoulli, 280
  de Bonferroni, 415
  de Boole, 415
  de números harmônicos, 272–273
  demonstração por indução matemática, 271
Inequação de Bernoulli, 280
Inequação de Bonferroni, 415
Inequação de Boole, 415
Inferência, regras de, 64–67, 105
  para correção de programa, 323–324
  para proposições, 71–72
Injeção, 135
Insertion sort, 172, 174, 257
  complexidade de caso médio de, 432–433
  complexidade de pior caso de, 196
Insertion sort binário, 172, 174, 179
Instalação controlada por três interruptores, circuito para, 764
Instalação controlada por três interruptores, circuito para, 764
Instanciação existencial, 70, 71
Instanciação universal, 70, 71
Instrumentos de busca, 734
Inteiro sem fator quadrático, 513
Inteligência artificial, 332
  em conjuntos difusos, 131–132
  em lógica fuzzy, 19
Intenção, de banco de dados, 531
Interior de um polígono simples, 288
International Standard Book Number (ISBN), 209
Internet
  busca, 13
  instrumentos de busca, 734
Internet Movie Database, 593
Internet Protocol (IP) de multicasting, 726
Interseção
  de conjuntos, 121–122, 126–128, 163
  de conjuntos fuzzy, 133
  de multiconjuntos, 132
Intervalo aberto, 282
Invariante para isomorfismo de grafo, 615, 676
Invariáveis, laços, 326–327, 329
Inverso
  de congruência de módulo m, 234
  de funções, 139–140, 163
  de implicação, 8, 104
  de matriz, 255
Inverso, multiplicativo, 54
Inversões, em permutação, número esperado de, 430–431
Inversor, 761, 781
Irmão de vértice, 686, 743
Irracionalidade de $\sqrt{2}$, 80–81, 292, 332
ISBN (International Standard Book Number), 209
Isobutano, 688
Iteração, 311
Iwaniec, Henryk, 215
Jarra, 633
Java, 135
Jogadores de futebol, 424
Jogo
  Chomp, 91
  da velha, 705
Josephus, Flavius, 459
Junção de matrizes zero-um, 252
Junção de relações n-árias, 532–534
Juros compostos, 451
Kaliningrad, Rússia, 633
Karnaugh, Maurice, 768
Kempe, Alfred Bray, 668
Kleene, Stephen Cole, 805, 817
Knuth, Donald E., 172, 182, 184, 189
Kruskal, Joseph Bernard, 739, 740
Kuratowski, Kazimierz, 663
Laços
  agrupados, 51–52
  dentro de laços, 858
  em grafos orientados, 541, 582
  invariáveis, 326–327, 329
Lados de um polígono, 288
Lady Byron, 27
Lagarias, Jeffrey, 102
Lagostas, 471
Lamé, Gabriel, 299
Landau, Edmund, 183
Laplace, Pierre Simon, 393, 394, 420
Largura de banda, 680
Leis de De Morgan
  demonstradas por indução matemática, 274–275
  para álgebra booleana, 753, 757
  para conjuntos, 124–126
  para proposições, 22, 25–26
  para quantificadores, 40–41
Leis do Pensamento, As (Boole), 3, 5, 420, 749
Lema, 75, 105
Liber Abaci (Fibonacci), 298
Liceu, 2
Ligando variáveis, 38–39
Limitante inferior
  de reticulado, 586
  de poset, 574, 582
Limitante superior, 846
  de poset, 574, 582
  de reticulados, 586
Limitante
  inferior
    de poset, 574, 582
    reticulado, 586
  maior inferior, 582
    de poset, 574
  menor superior, 582, 846
    de poset, 574
  superior, 846
    de poset, 574, 582
    reticulado, 586
Limite, definição de, 55
Linearidade da esperança, 429–431, 442
Linguagem, 785–793, 838
  formal, 786
  gerada pela gramática, 788
  livre de contexto, 789
  natural, 785–786
  reconhecido por autômato de estado finito, 806–814, 839
  regular, 789
Linguagem de consulta para banco de dados, 535–536
Linguagem de programação C, 346, 559
Linguagem formal, 786
Linguagem livre de contexto, 789
Linguagem natural, 785–786
Linguagem reconhecida, 806–814, 839
Linguagem regular, 789
Linguagens de programação, 68, 135, 346, 559
Linguagens de programação computacional, 68, 135
Linguagens reconhecidas, 806
Linha de matriz, 247
Linhas telefônicas
  rede de computadores com links de comunicações de mão única, 591

rede de computadores com links de diagnóstico, 591
rede de computadores com links múltiplos de mão única, 592
Listas, unindo duas, 319
Listas de adjacência, 611
Literal, 758, 781
Littlewood, John E., 93
Logaritmo iterado, 311
Lógica, 1–58, 104–105
fuzzy, 19
proposicional, 2
regras de inferência para, 64–67
Lógica fuzzy, 19
Lógica proposicional, 2
Lógica simbólica (Carroll), 44
Lógica simbólica (Venn), 115
Loteria, 394–395
Lovelace, Condessa de (Augusta Ada), 25, 27
Lucas, Edouard, 452
Lucky numbers, 518
Lukasiewicz, Jan, 719, 744
MAD Magazine, 184
Maior elemento
complexidade tempo de encontro, 194
do poset, 573, 582
em seqüência finita, 169
Maior limitante inferior, 582
de poset, 574
Maior limite de somatória, 153
Mãos de cartas, 358–359, 376–377
Mãos de pôquer, 358–359
Mapa de Karnaugh, 768–774, 781
Mapas K, 768–774, 781
Mapas
coloração de, 666–667
Karnaugh, 768–774
rodoviários, 595
Máquina de atraso, 799–800
Máquina de atraso unitário, 799–800
Máquina de Turing não determinística, 832
Máquina de Moore, 801
Máquina de venda, 797–798
Máquinas
de atraso unitário, 799–800
de estados finitos, 785, 798, 839
com saídas, 798–801
sem saída, 804–814
Mealy, 801
Moore, 801
Turing, 785, 825, 826, 827–832, 839
conjuntos reconhecidos por, 830–831
definição de, 828–830
funções computacionais com, 831–832
não determinística, 832
tipos de, 832
venda, 797–798
Máquinas de estado finito, 785, 796–814, 839
com saídas, 798–801
extensão de função de transição em, 805
sem saída, 804–814
Máquinas Mealy, 801, 839
Máquinas de Turing, 785, 825, 826, 827–832, 839
conjuntos reconhecidos por, 830–831
definição de, 828–830
em complexidade computacional, 833
funções de computação com, 831–832
não determinística, 832
tipos de, 832
Margem do tabuleiro de xadrez, 97–102
Maridos, ciumentos, 632–633
Markup languages, 796
Master Theorem, 479
Matemática discreta, definição de, vii, viii
Mathematical Analysis of Logic,The (Boole), 5
Matriz linear, 606, 607
Matriz(es), 246–254
adição de, 247–248
adjacência, 612–614, 676
contando caminhos entre vértices por, 628–629
coluna de, 247
definição de, 257
diagonal, 255
esparsa, 614
identidade, 251, 257
incidência, 614–615, 676
inversa de, 255
inversível, 255
linha de, 247
multiplicação de, 249–250
algoritmos para, 249–250
rápida, 476–478
potências de, 251–252
produto Booleano de, 252–254, 257
quadrada, 251
representação de relações usando, 538–540
simétrica, 251, 257
transposta de, 251, 257
triangular, 260
zero-um, 252–254, 257
algoritmo de Warshall e, 550–553
conjunção de, 252
de fecho transitivo, 549–550
encontro de, 252
produto Booleano de, 252–254
representação de relações utilizando, 538–540
Matriz bidimensional, 606–607
Matriz diagonal, 255
Matriz esparsa, 614
Matriz identidade, 251, 257
Matriz inversível, 255
Matriz ou tabela
bidimensional, 606
linear, 606, 607
Matriz quadrada, 251
Matriz simétrica, 251, 257
Matriz triangular, 260
Matrizes de adjacências, 612–614, 677
contando caminhos entre vértices por, 628–629
Matrizes de incidência, 614–615, 676
Matrizes zero-um, 252–254, 257
algoritmo de Warshall e, 550–553
conjunção de, 252
do fecho transitivo, 549–550
encontro de, 252
produto Booleano de, 252–254
representação de relações usando, 538–540
Maurolico, Francesco, 265
Máximo de uma seqüência, 475
Máximo divisor comum (mdc), 215–217, 232–234, 258
Maxtermo, 760
McCarthy, John, 332
McCluskey, Edward J., 775
mdc (máximo divisor comum), 215–217, 232–234, 258
Mealy, G. H., 801
Média, 178
aritmética, 95, 282
desvio da, 439
geométrica, 95, 282
harmônica, 103
quadrática, 103
Média aritmética, 95, 282
Média geométrico, 95, 282
Média harmônica, 103
Média quadrática, 103
Médio, 178
Meias, 353
Meigu, Guan, 638
Meio somador, 764, 781
Meio subtrator, 766
Menor elemento de poset, 573, 582
Menor limitante superior, 582, 846
de poset, 574
Menor limite de somatória, 153
Merge sort, 172, 317–321, 329, 475
complexidade de, 479
recursiva, 318
Merge sort recursiva, 318
Mersenne, Marin, 212, 213
Mesh de árvores, 748
Mesma ordem, 182
Metacaracteres, 796
Metafont, 184
Método congruente linear, 206
Método de Horner, 199
Método de Quine-McCluskey, 768, 775–779
Método iterativo, para fatoriais, 317
Método probabilístico, 413–414, 442
Método(s)
congruente linear, 206
Critical Path, 587
de Horner, 199
probabilístico, 413–414, 442
prova, 75–102, 105
Quine-McCluskey, 767, 775–779
Millbanke, Annabella, 27
Minimização
de circuitos combinatórios, 766–779
de funções Booleanas, 766–779, 781
Minimização de máquina, 810
Mínimo, de seqüência, 475
Mínimo múltiplo comum (MMC), 216–217, 257
Mintermo, 758, 781
mod, 202, 203, 205–207, 225–226
Moda, 178

Modelagem
  com árvores, 688–690
  com grafos, 592–595
  com relações de recorrência, 450–456
  computacional, 785–837
Modelo relacional de dados, 531–532, 581
Modelo reticulado do fluxo de informação, 575–576
Modo de transferência assíncrona (ATM), 144
Módulo, 206
Modus ponens, 65, 66
Modus tollens, 66
Moeda de cinco centavos, 175–176
Moeda de um centavo, 175–176
Moedas, 175–176
Moedas, troco usando o menor número, 175–176
Moléculas, modelagem com árvores, 688
Monociclo, 750
Montmort, Pierre Raymond de, 511, 518
Moore, E. F., 801
Multiconjuntos, 132
Multigrafo orientado, 591, 675
  caminhos em, 623
Multigrafos, 590, 675
  caminho euleriano, 637
  ciclo euleriano, 637
  não orientado, 592
Multiplexer, 766
Multiplicação
  de funções, 135–136
  de matrizes, 249–250
    algoritmos para, 249–250
    rápida, 476–478
  de números inteiros
    algoritmos para, 224–226
    rápida, 475–476
  encadeada de matrizes, 250
Multiplicação encadeada de matriz, 250
Multiplicação rápida
  de matrizes, 476–478
  de números inteiros, 475–476
Multiplicador, 206
Multiplicidade de elementos em multiconjuntos, 132
Múltiplo
  de números inteiros, 201
  mínimo comum, 215–217, 257
Múltiplo comum, mínimo, 216, 217
m-upla, 533
Namagiri, 94
Não rotulada
  caixas, 376
  objetos, 376
Não terminais, 787
Não-conformistas, 420
Naur, Peter, 792
NAUTY, 618
Naval Ordnance Lab, 811
Navy WAVES, 811
n-cubos, 602
NE, 29, 759, 781
Necessário, 6
  e suficiente, 9
Negação
  de proposições, 3–4, 104
  de quantificação existencial, 40
  de quantificação universal, 40
  de quantificadores agrupados, 57–58
Negativa
  falsa, 419
  verdadeira, 419
Negativo verdadeiro, 419
Netuno, 424
New Foundations of Mathematical Logic, (Quine), 776
New Jersey Crags, 154
Nim, jogo do, 704–705, 707
Níveis de prioridade, 11, 38
Nível de vértice, 691, 743
Nós, 541, 589
NOT, 13
Notação
  complemento de dois, 230–231
  complemento de um, 230
  construção do conjunto, 112
  dependência, 784
  infixa, 719–721, 744
  o pequeno, 192
  O grande, 180–186
  Omega grande, 180, 189–190
  para produtos, 162
    fórmula bem formada em, 724
  polonesa, 719, 744
  polonesa reversa, 720, 744
  pós-fixa, 719–721, 744
  prefixa, 719–721, 744
  somatória, 153
  Teta grande, 180, 189–190
Notação de conjunto, com quantificadores, 119
Notação de dependência, 784
Notação de produto, 162
Notação de somatório, 153
Notação decimal, 168, 219
Notação infixa, 719–721, 744
Notação *O* grande, 180–186
Notação *o* pequeno, 192
Notação Ômega grande, 182, 189–190
Notação polonesa, 719, 744
Notação polonesa reversa, 720, 744
Notação pós-fixa, 719–721, 744
Notação prefixa, 719–721, 744
Notação Teta grande, 182, 189–190
NOU, 29, 759, 781
Nova, 101
NP, classe, 197, 836
Número cromático, 668, 676
  aresta, 674
  de grafo, 667
  k-crítico, 674
Número cromático de arestas, 674
Número de Bacon, 623
Número de cruzamentos, 666
Número de Erdös, 584, 623, 624
Número de hospedagem (hostid), 341
Número de independência, 680
Número de livro de um grafo, 682
Número de Ramsey, 352
Número de telefone, 594
Número de transmissão (*netid*), 341
Número inteiro ímpar, 77
Número inteiro par, 77
Número primo relativo mutuamente, 260
Número telefônico, 337
Número(s) harmônico(s), 192
  inequação de, 272–273
Número(s)
  Bacon, 623
  Bell, 566
  Carmichael, 240–241, 257
  Catalan, 456
  cromático, 667, 668, 676
    arestas, 674
  cruzamentos, 666
  Cunningham, 214
  de Ulam, 165
  Erdös, 584, 623, 624
  Fibonacci, 297–298, 316–317, 709
    algoritmo de iteração para, 317
    algoritmos recursivos para, 316
    coelhos e, 451–452
    fórmula para, 463–464
  grandes, lei dos, 407
  harmônico, 192
    inequação de, 272–273
  independência, 680
  irracional, 217
  Lucas, 330, 471
  lucky, 518
  natural, 113
  os dez mais procurados, 214
  perfeito, 218
  pseudoaleatório, 207
  racional, 105, 113
    contabilidade de, 158, 163
  Ramsey, 352
  real, 113, 845 –849
  Stirling, de segunda ordem, 378
Números "Dez Mais Procurados", 214
Números de Bell, 566
Números de Carmichael, 240–241, 257
Números de Catalan, 456
Números de Cunningham, 214
Números de Fibonacci, 297
  algoritmo de iteração para, 317
  algoritmos recursivos para, 316
  coelhos e, 451–452
  e código de Huffman, 709
  fórmula para, 463–464
Números de Lucas, 330, 471
Números de Ulam, 165
Números inteiros, 112
  adição de, algoritmos para, 222–224
  aplicações com, 231–244
  axiomas para, 849
  composto, 210, 211, 217, 257
  conjunto dos, 112
  divisão de, 200–208
  encontrar o maior elemento em uma seqüência finita de, 170
  grandes, aritmética da computação com, 237–238
  ímpar, 77
  máximo divisor comum de, 215–217
  mínimo múltiplo comum de, 216, 217
  multiplicação de
    algoritmos para, 224–226
    rápida, 475–476

par, 77
partição de, 310
perfeito, 218
pseudoprimos, 238–241, 245, 257
representação de, 219–222, 230–231
sem fator quadrático, 513
Números inteiros compostos, 189–190, 210, 211, 217, 257
Números inteiros compostos consecutivos, 217
Números inteiros positivos, 113
axiomas para, 849
Números irracionais, 182
Números naturais, 113
Números primos, 210–215
crivo de Eratóstenes e, 210
definição de, 257
distribuição de, 213
gêmeos, 215
infinitos, 212
Mersenne, 212, 257
pseudoprimos, 238–241, 245, 257
relativamente, 216, 257
relativos mutualmente, 260
Números primos entre si, 216, 257
Números primos gêmeos, 215
Números pseudo-aleatórios, 206–207
Números racionais, 105, 113
contabilidade de, 159, 163, 218
Números racionais positivos, contabilidade de, 158
Números reais, 113
construção, 849
maior limitante, 846
menor limitante, 846
não contabilidade de, 160
Números Stirling de segundo tipo, 378
n-upla, 531–532
ordenada, 117
n-uplas ordenadas, 117
Objeto(s), 111–112
distinto, 376
e caixas distintas, 376–377
e caixas idênticas, 377–378
idêntico, 376
e caixas idênticas, 377, 378–379
não rotulado, 376
Odlyzko, Andrew, 154
On-Line Encyclopedia of Integer Sequences (Sloane e Plouffe), 153
Operações
bit, 15
em conjuntos, 121–130
em relações n-árias, 532–535
número estimado de, de algoritmos, 180–182
número inteiro, algoritmos para, 222–226
Operações binárias, 15, 104
Operador de negação, 4
Operador de seleção, 532, 582
Operadores binários, 15–16, 104
Operadores lógicos, 104
funcionalmente completos, 29
precedência de, 10–11
Operadores
bitwise, 15–16, 104
conjunto funcionalmente completo de, para proposições, 29
fórmula bem formada de, 301
lógicos, 104
funcionalmente completos, 29
prioridade de, 10–11
negação, 4
seleção, 532
Operantes, fórmula bem formada de, 301
Ópio, 424
Oracle of Bacon, 624
Ordem, 190
de quantificadores, 52–54
mesma, 182
Ordem alfabética, 382
Ordem de nível de vértice, 745
Ordem de torneio *(tournament sort)*, 172, 708
Ordem lexicográfica, 307, 382–384, 385, 569–570
Ordem linear, 568, 582
Ordem pecking, 680
Ordem total compatível, 576, 582
Ordenação rápida, 172, 322
Ordenação topológica, 576–578, 582
Ordenação
bubble sort, 173, 257
bidirecional, 259
complexidade de pior caso de, 194–195
insertion sort, 172, 174, 257
complexidade de caso médio, 432–433
complexidade de pior caso de, 196
merge sort, 172, 317–321, 329, 475
complexidade da, 479
recursiva, 318
por inserção binária, 172, 179
por seleção (selection sort), 172, 178–179
quick sort, 172, 322
shaker sort, 172, 259
torneio (tournament sort), 172, 708
Ordenamento
alfabético, 382
lexicográfico, 307, 382–384, 569–570
linear, 568
parcial, 566–578, 582
quase-ordem, 585
total, 568, 582
Ordenamento total, 567, 582
compatível, 576, 582
Ordens parciais, 566–578, 582
ordem total compatível com, 582
Ore, Oystein, 641
Orelhas não sobrepostas, 292
Orientação de um grafo não orientado, 679
OU (OR), 13, 15–16
Ou exclusivo, 5, 104
Ou inclusivo, 5
P, classe, 197, 836
P(n,r), 386
Página de busca da Web, 13
Páginas da Web, 13, 595, 627, 734
Pai de vértice, 686, 743
Palavra, 787
Palíndromo, 178, 346, 795
Par
do diabo, 621
ordenado, 117
Paradoxo, 112
de Russell, 121
de São Petersburgo, 444
do barbeiro, 19
Parâmetros, 181
Parênteses, cadeias balanceadas de, 332
Pares ordenados, 117, 120
Pares relativamente primos, 216, 257
Partição, 499
de conjunto, 560–562, 582
de número inteiro positivo, 310, 379
refinamento de, 565
Pascal, 135
Pascal, Blaise, 239, 366, 393, 400
Passeio fechado, 622
Passo base, 264, 284, 299, 328, 690, 706
Passo de indução, 264, 690, 706
Passo recursivo, 299
Pasta vazia, 114
Pearl Harbor, 811
Peirce, Charles Sanders, 29, 32
Pelc, A., 483
Pequeno Teorema de Fermat, 239, 258
Percurso em árvores, 710–722, 743
em ordem simétrica, 712, 714, 715, 718, 743
pós-ordem, 712, 714–716, 717, 718, 744
pré-ordem, 712–713, 718, 743
Percurso em ordem simétrica, 712, 713, 714, 715, 716, 718, 743
Percurso em pós-ordem, 712, 715–716, 717, 718, 744
Percurso em pré-ordem, 712–713, 718, 743
Perfeito
número, 218
números inteiros, 218
potência, 87
Peripatético, 2
Permutação aleatória, geradora, 445
Permutação caótica, 510–512, 514
número de, 510, 514
Permutações, 355–357, 386
aleatórias, 445
com objetos idênticos, 375–376
com repetição, 371
gerando, 382–384
inversões em, número esperado de, 430–431
Permutações generalizadas, 370–371
PERT (Program Evaluation and ReviewTechnique), 587
Pertence, 112
Pertinência, grau de, 132
Petersen, Julius Peter Christian, 646
Platão, 2
Plouffe, Simon, 154
Poda alpha-beta, 707, 748
Polígono, 288
com orelhas não sobrepostas, 292
convexo, 288
diagonal de, 288
exterior de, 288

interior de, 288
lados de, 288
não convexo, 288
simples, 288
vértices de, 288
Polígono convexo, 288
Polígono simples, 288
triangulação de, 288
Poliominós, 99, 277
Pólya, George, 102
Ponte, em grafos, 625
Ponteiro, posição do, representação digital da, 642–643
Ponto reticulado, 330
Pontos de articulação, 625
Porta *E*, 761, 781
Porta *OU*, 761, 781
Porta *threshold*, 783
Portas, lógicas, 760–765
combinação de, 761–763
E, 761, 781
NE, 766
NOU, 766
OU, 761, 781
threshold, 783
Portas lógicas, 760–765
combinação de, 761–763
E, 761, 781
NE, 766
NOU, 766
OU, 761, 781
threshold, 783
Portas *NE*, 766
Portas *NOU*, 766
Pós-condição, 33, 323
Poset, 566, 582
anticadeia em, 585
bem fundado, 581
cadeia em, 585
denso, 581
diagrama de Hasse, 571–572, 573, 582
dual de, 579
elemento maximal de, 572, 582
elemento minimal de, 572, 573, 582
elementos comparáveis em, 567, 582
elementos incomparáveis em, 567, 582
maior elemento de, 573, 582
maior limitante de, 574, 582
menor elemento de, 573, 582
menor limitante de, 574, 582
Poset bem fundado, 581
Posição inicial de uma máquina Turing, 829
Positivo
falso, 419
verdadeiro, 419
Postulados, 75
Potência, perfeita, 87
Potência booleana (de matrizes zero-um), 252–254
Potenciação, modular, 226–227
recursiva, 313
Potências
de matrizes, 251–252
de relação, 527, 581
Precondição, 33, 323
Predicado, 30, 31
Predicado nominal, 786
Predicado verbal, 786
Pré-imagem, 147, 163
Prêmio, 704
Premissa, 6, 44, 63
Pretendentes, 179, 293
ideal para, 293
Pretendentes, as, 179
Prim, Robert Clay, 738, 740
Primeira diferença, 460
Primeira diferença progressiva, 516
Primeiro buraco, 279
Primos de Mersenne, 212, 257
Princesa dos Paralelogramos, A, 27
Principia Mathematica (Whitehead e Russell), 28, 114
Princípio(s)
casa dos pombos, da, 347–353, 387
aplicações com, 351–353
generalização, do, 349–351, 387
de contagem, 335–340
de flutuação, 848
de inclusão–exclusão, 122, 341–343, 500–504, 514
aplicações com, 505–512
forma alternativa de, 506–507
de indução bem fundada, 585
de indução bem ordenada, 568–569
de indução matemática, 328
dualidade, para identidades booleanas, 754
Princípio da casa dos pombos, 347–353, 387
aplicações com, 351–353
generalização do, 349–351, 387
Princípio da casa dos pombos generalizado, 349–351, 387
Princípio da dualidade, para identidades booleanas, 754
Princípio da inclusão–exclusão, 122, 341–343, 500–504, 514
aplicações com, 505–512
forma alternativa de, 506–507
Princípio das gavetas de Dirichlet, 348
Probabilidade, discreta, 393–447
condicional, 400, 404–405, 442
de colisão em funções de hashing, 410–411
de combinatórias em eventos, 396–398, 403–404
definição de Laplace de, 394
determinando, 401–403
em resultados de testes médicos, 419–420
finita, 394–396
Probabilidade condicional, 400, 404–405, 442
Probabilidade discreta, 393–447, 442
condicional, 400, 404–405, 442
de colisão em funções de hashing, 410–411
de combinações de eventos, 396–398, 403–404
definição de Laplace de, 393
determinando, 401–403
finita, 394–396
Probabilidade finita, 394–396
Problema Collatz, 101–102
Problema da celebridade, 282
Problema da fofoca, por indução matemática, 282
Problema das casas e dos serviços públicos, 657–659
Problema das *n* rainhas, 731–732
Problema das Pontes de Königsberg, 633–634, 636, 638
Problema das três portas de Monty Hall, 398, 425
Problema de Josefo, 459
Problema de Kakutani, 102
Problema de lógica, 74
Problema de Newton-Pepys, 447
Problema de parada, 176–177, 834–835
Problema de satisfabilidade, máxima, 445
Problema de Siracusa, 102
Problema de travessia do rio, 632
Problema de Ulam, 102, 483
Problema do beijo, 154
Problema do caixeiro-viajante, 641, 649, 653–655, 676
Problema do carteiro chinês, 638, 682
Problema do cavalo, 647
com várias entradas, 647
Problema do chapéu, 430, 510
Problema do Par de Pontos Mais Próximo, 479–482
Problema do passeio de cavalo em um tabuleiro, 647
Problema dos maridos ciumentos, 632–633
Problema não passível de ser solucionado, 197
Problema P = NP, 836
Problema passível de ser solucionado, 197
Problema tratável, 197, 836
Problema(s)
aberto, 214–215
aniversário, 409–411
caixeiro-viajante, 641, 649, 653–655, 676
carteiro chinês, 638
celebridade, 282
chapéu, valor esperado, 430, 510
classes de, 835
Collatz, 102
da parada, 176–177, 834–835
das casas e dos serviços públicos, 657–659
de caminho mais curto, 647–655, 676
de Josefo, 459
de Kakutani, 102
de Newton-Pepys, 447
de NP-completo, 197, 655, 768
de parada, 835
de Siracusa, 102
de Ulam,102
décimo de Hilbert, 835
decisão, 834
dos maridos ciumentos, 632–633
galeria de arte, 675
insolúvel, 197, 834, 835
intratável, 197, 836
kissing (do beijo), 154

lógica, 74
máxima satisfabilidade, 445
Milênio, 836
n rainhas, 731–732
otimização, 174–176
P = NP, 836
Par Mais Próximo, 479–482
polinomiais no tempo, 836
pontes, 633–634, 636, 638
programar algoritmo voraz para, 179
exames finais, 671, 672
sim-ou-não, 834
solúvel, 197, 834
tratável, 197, 836
Problemas, Milênio, 837
Problemas abertos, 214–215
Problemas de agendamento, 179
algoritmo voraz para, 275–276
exames finais, 671, 672
projetos de software, 581
tarefas, 576–578
Problemas de caminho mais curto, 647–655, 676
Problemas de decisão, 834
Problemas de máxima satisfabilidade, 445
Problemas de NP completo, 197, 655, 768, 839
Problemas de otimização , 174–176
Problemas intratáveis, 197, 836
Problemas polinomiais de tempo, 836
Problemas sim-ou-não, 834
Problème de rencontres, 511, 518
Problème des ménages, 518
Procedimentos paralelos conectados em árvore, 689–690
Processamento
concomitante, 595
paralelo, 199, 606
conectado em árvores, 689–690
Produto
booleano, 749–750, 751, 757–759, 781
de matrizes zero-um, 252–254, 257
cartesiano, 117–118
Produtos cartesianos, 117–118
Profundidade de circuito combinatório, 766
Program Evaluation and Review Technique (PERT), 587
Programação de computadores, 792
Programação lógica, 45–46
Programas CAD, 772
Progressão(s)
aritmética, 151
geométrica, 150, 152
somas de, 270–271
Projeção de relações *n*-árias, 533, 582
Projeto do Número de Erdös, 623
Prolog, 45
Proposição **"se, então"**, 6, 8
Proposição "se e somente se ", 10
Proposição "somente se", 9
Proposição condicional, 6
para correção de programa, 324–326
Proposição(ões), 1–6, 74
composta, 3, 10–11, 21, 24, 104
consistente, 104
dual de, 29
forma normal disjuntiva de, 29
fórmula bem formada para, 301
satisfatória, 30
conjunção de, 4–5, 104
conjunto funcionalmente completo de operador para, 29
definição de, 1, 104
disjunção de, 4–5, 104
equivalente, 8, 22–30, 104
negação de, 3–4, 104
regras de inferência para, 64–67
valor de verdade de, 3
Proposições compostas, 3, 10–11, 21, 22, 104
consistentes, 104
dual de, 29
equivalências de, em álgebra booleana, 750
forma normal disjuntiva de, 29
fórmula bem formada para, 301
satisfatória, 30
Proposições compostas consistentes, 104
Proposições compostas satisfatórias, 30
Proposições equivalentes, 8, 21–30
Propriedade
absorção
para álgebra booleana, 753, 754
para conjuntos, 124
para proposições, 24
para reticulados, 585
arquimediana, 849
associativa
da adição, 845
da multiplicação, 845
para álgebra booleana, 753, 755
para conjuntos, 124
para proposições, 24
para reticulados, 585
complementares, para conjuntos, 124
comutativa
da adição, 845
da multiplicação, 845
para álgebra booleana, 753, 755
para conjuntos, 124
para proposições, 26
para reticulados, 585
compatibilidade
de adição, 846
de multiplicação, 846
da boa ordenação, 290–291, 328, 845
Propriedade arquimediana, 849
Propriedade da boa ordenação, 290–291, 328,
Propriedade da complementação, para conjuntos, 124
Propriedade da dupla negação, para proposições, 26
Propriedade da Tricotomia, 846
Propriedade da unidade, em álgebra booleana, 753
Propriedade de Compatibilidade da Adição, 846
Propriedade do elemento inverso
da adição, 846
da multiplicação, 846
Propriedade do zero, em álgebra booleana, 753
Propriedade dos elementos neutros, 845
Propriedade modulares, em álgebra Booleana, 757
Propriedade monótona crescente de um grafo, 681
Propriedade monótona decrescente de um grafo, 681
Propriedade transitiva, 846
da completude, 846
de complementação, para conjuntos, 124
de De Morgan
para álgebra booleana, 753, 757
para conjuntos, 124
para proposições, 22, 25–26
para quantificadores, 40–41
prova por indução matemática, 274–275
de destacamento, 65
de duplo complemento, em álgebra booleana, 753, 755
idempotente
para álgebra booleana, 753, 757
para conjuntos, 124
para proposições, 24
para reticulados, 585
de fechamento
para adição, 845
para multiplicação, 845
de grandes números, 407
de indução matemática, 849
distributiva, 846
para álgebra booleana, 752, 753, 755
para conjuntos, 124
para proposições, 23, 24
dominação
para álgebra booleana, 753
para conjuntos, 124
para proposições, 27
identidade
de adição, 845
de multiplicação, 845
para álgebra booleana, 752–754, 755
para conjuntos, 124
para proposições, 26
inversa
da adição, 846
da multiplicação, 846
negação, para proposições, 26
transitiva, 846
tricotomia, 846
utilizada para demonstrar fatos básicos, 846–849
Propriedades associativas
da adição, 845
da multiplicação, 845
para álgebra booleana, 753, 755
para conjuntos, 124
para proposições, 24
para reticulados, 585
Propriedades comutativas
da adição, 845
da multiplicação, 845
para álgebra booleana, 753, 755
para conjuntos, 124
para proposições, 26
para reticulados, 585

Propriedades de absorção
para a álgebra booleana, 753, 754
para conjuntos, 124
para proposições, 24
para reticulados, 585
Propriedades de compatibilidade, 846
Propriedades de dominação
para álgebra booleana, 753
para conjuntos, 124
para proposições, 27
Propriedades de fechamento
para a adição, 845
para a multiplicação, 845
Propriedades de negação, para proposições, 24
Propriedades distributivas
para álgebra booleana, 752, 753, 755
para conjuntos, 124
para proposições, 23, 24
Propriedades dos complementares, para conjuntos, 124
Propriedades dos elementos neutros
da adição, 845
da multiplicação, 845
para álgebra booleana, 752–754, 755
para conjuntos, 124
para proposições, 26
Propriedades idempotentes
para álgebra booleana, 753, 757
para conjuntos, 124
para proposições, 24
para reticulados, 585
Protestantes não-conformistas, 420
Prova FBF, O Jogo da Lógica (Allen), 74
Pseudocódigo, 169, 854–859
Pseudografo, 590, 592, 675
Pseudoprimo, 238–239, 245, 257
Pula-pula motorizado, 750
Quantificação existencial, 36, 105
negação de, 39–41
Quantificação nula, 49
Quantificação universal, 34, 105
negação de, 39–41
Quantificação
como laços, 51–52
existencial, 36, 105
negação de, 39
nula, 49
universal, 34, 105
negação de, 39
Quantificador de unicidade, 37
Quantificador existencial, 36–37
Quantificador universal, 34–36
Quantificadores, 34–37
agrupados, 50–58
negando, 57–58
com domínio restrito, 38
conjunto-verdade de, 119
de unicidade, 37
domínio de, 34
existencial, 36–37
lei De Morgan para, 40–41
negando, 39–41
ordem dos, 52–54
prioridade de, 38
regras de inferência para, 70
universal, 34–36
usando em especificações de sistema, 43–44
usando em notação de conjunto com, 119
Quantificadores agrupados, 50–58
negação, 57–58
Quartos, 175–176
Quase ordem, 585
Quebra de Sigilo (filme), 242
Quebra-cabeça, 292
Quebra-cabeça das crianças enlameadas, 14–15
Quebra-cabeças lógicos, 13–15
Queijo, no Reve's puzzle, 454
Quine, Willard van Orman, 775, 776
Quociente, 202, 850
Autômato, 817
r-permutação, 355, 386
Raças de cavalo, 27
Raciocínio
circular, 84, 105
dedutiva, 264
direto, 94
indutivo, 264
para trás, 94
probabilístico, 398
Raciocínio circular, 84, 105
Raciocínio direto, 94
Raciocínio direto e para trás, 94
Raciocínio probabilístico, 398
Rado, Tibor, 838
Rainhas em um tabuleiro de xadrez, 679
Raio de um grafo, 680
Raízes características, 461
Ramanujan, Srinivasa, 93, 94
Ramaré, O., 215
Ramsey, Frank Plumpton, 352
Razão comum, 150
r-combinação, 357, 386
Rebelião jacobina, 145
Recobrir, 579
Reconhecimento de linguagem, 801
Recordes mundiais, para números primos gêmeos, 215
Recursão, 294
Rede
conectada por árvore, 689–690
de computadores, 590
com links de comunicações de mão única, 591
com links de diagnóstico, 591
com links múltiplos, 590
com links múltiplos de mão única, 592
de áreas locais, 605–606
de distribuição múltipla, 726
de interconexão para computação paralela, 606–607
porta, 761–763
exemplos de, 763–764
minimização de, 766–779
profundidade de, 766
Rede de malha, 606–607
Rede de telefones, 594, 625
Rede de telefones de longa distância, 594
Redes de área local, 605–606
Redes de interconexão para computação paralela, 606–607
Redes de portas, 761–763
exemplos de, 763–764
minimização de, 766–779
profundidade de, 766
Refeição chinesa, preparo, 585
Refinamento, de partição, 565
Regiões de grafos de representação planar, 659, 676
Registrador *Shift*, 784
Registros, 531
Registros de índice, 672
Regra da simplificação de inferência, 66
Regra da soma, 338, 386–387
Regra de composição, 324
Regra de conjunção de inferência, 66
Regra de exclusão, 299
Regra de inferência de adição, 66
Regra de produção na gramática, 787
Regra do produto, 336
Regras de inferência, 63–67, 70, 105
para correção de programa, 323–324
para proposições, 64–67
para quantificadores, 70
para sentenças quantificadas, 71–72
Regras Prolog, 45
Relação "maior que ou igual a", 566–567
Relação anti-simétrica, 523–524, 581
representações
usando dígrafos, 541
usando matrizes, 538–539
Relação transitiva, 524–525, 527, 581
representação, usando dígrafos, 542
Relação(ões), 519–587
"maior ou igual a," 566–567
"menor que ou igual a," 567
anti-simétrica, 523–524, 581
assimétrica, 528
binária, 519, 581
caminhos em, 546–547
circular, 584
combinando, 525–527
complementar, 528
composta de, 526–527, 581
conectividade, 547, 582
contagem, 522
de recobrimento, 579
diagonal, 545
divisibilidade, 567
domínios de, 530
em conjunto, 118, 521–522
equivalência, 555–562, 582
fecho de, 544–553, 582
reflexivo, 544, 582
simétrico, 545, 582
transitivo, 544, 547–550, 582
de computação, 550–553
funções como, 520
inclusão, 567
inversa, 528, 581
irreflexiva, 528
n-ária, 530–536, 581
domínio de, 531
operações em, 532–535

potências de, 527
progenitor, 526
propriedades de, 522–525
recorrência de divisão e conquista, 474
reflexiva, 522, 581
representação
usando dígrafos, 541–542
usando matrizes, 538–540
simétrica, 523–524, 582
representação
usando dígrafos, 541, 542
usando matrizes, 538
transitiva, 524–525, 527, 582
representação, usando dígrafos, 542
Relações de equivalência, 555–562, 582
definição de, 555
Relações de recorrência, 449–460
condição inicial para, 450
definição de, 450, 513
divisão e resolução, 474–482
homogêneas associadas, 467
linear e heterogênea, 467–471, 514
linear e homogênea, 460, 461–467, 513
modelos com, 450–456
resolvendo, 460–471
funções generalizadas para, 493–495
simultânea, 472
Relações de recorrência associadas e homogêneas, 467
Relações de recorrência lineares e heterogêneas, 467–471, 514
Relações de recorrência lineares e homogêneas, 460, 461–467, 513
Relações *n*-árias, 530–536, 581
domínio de, 530
operações em, 532–535
Rencontres (combinações), 511, 518
Repetição
combinações com, 371–375
permutações com, 371
Repolho, 632
Reposição
amostra com, 396
amostra sem, 396
Representação computacional, de conjuntos, 129–130
Representações unárias, 831
Representações
binária, 219
complemento de um, 230
decimais, 168, 219
hexadecimais, 219–220
octais, 220
unária, 831
Representativa, de classe de equivalência, 558
Resíduo quadrático, 246
Resolução, 66, 68
Resolver um problema de decisão, 836
Resto, 202, 203, 257
Resultados, 75
Reticulados, 574–576, 582
complementado, 586, 755
conjunção em, 585
distributivo, 586, 755
dualidade em, 587
encontro em, 585
limitado, 586
modular, 587
propriedades associativas para, 585
propriedades comutativas para, 585
propriedades de absorção para, 585
propriedades idempotentes para, 585
Reticulados limitados, 586
Reve's puzzle, 454
Reverso de grafo orientado, 611
Rivest, Ronald, 241, 242
Rodas, 601, 676
Rook polynomials, 518
Roy, Bernard, 551
Run, 440
Russell, Bertrand, 112, 114
Saídas
máquinas de estado finito com, 798–801
máquinas de estado finito sem, 804–814
Samos, Michael, 480
Sânscrito, 792
Satélite espacial, 424
Screw, Arquimedes, 848
Segredos, 266
Segundo princípio de indução matemática, 284
Seis graus de Separação (Guare), 623
Selection sort, 172, 178–179
Sem perda de generalidade, 89
Semântica, 785
Senhas, 60, 340
Sentença, 786, 787
Sentenças em português
traduzindo expressões lógicas para, 42–43
traduzindo para expressões lógicas, 11–12, 42–43, 56–57
Seqüência autogeradora de Golomb, 333
Seqüência(s), 150–153
autogeradora, 333
de Bruijn, 682
definição de, 150
do algoritmo recursivo, 312, 313
de números inteiros, 151–153
do algoritmo recursivo, 312, 313
encontrar o máximo e o mínimo de, 475
estritamente crescente, 351
estritamente decrescente, 351
funções geradoras para, 484
distinguíveis, 827
indistinguíveis, 827
nula, 151, 787
ou cadeias de bit, 15, 104, 129-130
Seqüenciamento de DNA, 638
distinguíveis, 827
Seqüências ou cadeias de bit, 15, 104, 129–130
com um número ímpar de 1s e terminando com pelo menos dois 0s, 809
contagem, 336
sem 0s consecutivos, 454–455
decodificação, 700–701
em comunicação interestelar, 424
extensão de, 15, 454–455
geração da maior, subseqüente, 385
Seqüências reconhecidas, 806
Séries de potências, 485–488
formais, 485
Séries
geométrica, 155
harmônica, 273
infinita, 157
potência, 485–488
formal, 485
Serpentes venenosas, 702
Shaker sort, 172, 259
Shamir, Adi, 241, 242
Shannon, Claude Elwood, 749, 750
Sheffer, Henry Maurice, 28, 29
Silogismo
disjuntivo, 66
hipotético, 66
Símbolo(s), 787
de Landau, 182
de término, 787
inicial, 787
Sintaxe, 785
Sistema de endereçamento global, 711
Sistema RSA, 241–244
codificação, 242–243
de decodificação, 243
Sistemas de arquivo de computador, 689
Sistemas de aviação, modelagem por grafos com pesos, 648
Sistemas de Endereçamento Globais, 711
Sistemas de especificações, 12, 43–44
consistentes, 12
usando quantificadores em, 43–44
Sistemas de hardware, 12
Sistemas de software, 12
S_k-árvore, 745
Sloane, Neil, 154
Smullyan, Raymond, 13, 14
Solução de uma relação de recorrência, 450
Soma booleana, 749–750, 751, 757–759, 781
Soma telescópica, 162
Soma(s)
booleana, 749–750, 751, 757–759, 781
de funções, 135–136
de matrizes, 247–248
de multiconjuntos, 132
de progressões geométricas, 270–271
de subconjuntos, 732
dos primeiros n números inteiros positivos, 157, 162, 267–268
Somador completo, 764, 781
Somador(es), 764–765
completo, 764, 781
meio, 764, 781
Somatória dupla, 156
Somatórios, 153–158
demonstração, por indução matemática, 267
duplas, 156
índice de, 153
maior limite de, 153
menor limite de, 153
Somerville, Mary, 27
Sorvedouro, 841
Spam, 421–423

Spiders, Web, 734
SQL, 535–536
Stirling, James, 145
Structured query language, 535–536
Subárvore, 686, 687, 743
Subárvore direita, 687
Subárvore esquerda, 687
Subconjunto estrito, 115, 163
Subconjuntos, 115, 163
  de conjunto finito
    contagem, 338
    número de, 273–274
  próprio, 163
  somas de, 732
Subdivisão elementar, 663, 676
Subgrafo, 607, 678
  induzido, 678
Subseqüência, 351
Substantivo, 786
Substituição, 224
Subtratores
  completos, 766
  meio, 766
Sucesso, 406
Sucessor
  de números inteiros, 849
  de conjunto, 132
Suficiente, 6
  necessária e, 9
Sun-Tzu, 235
Tabela de estados para máquina de estado-finito, 798
Tabelas
  de estados, 798, 802
  de pertinência, 126, 163
  verdade, 3
Tabela-verdade, 3–6, 10, 104
  para conjunção de duas proposições, 4
  para disjunção de duas proposições, 4
  para equivalências lógicas, 24
  para implicação, 6
  para negação de uma proposição, 3
  para ou exclusivo de duas proposições, 6
  por proposição bicondicional, 9
Tabuleiro, 277
  passeio de cavalo em, 647
  problema das *n* rainhas, 731–732
  quadrados controlados pela rainha, 679
Tabuleiro de xadrez, ladrilhamento de, 97–102, 277, 283
  ideal, 293
Tarefa estável, 179, 293
Tarefas, atribuição de, 604
Tarefas, freqüência, 672
Tautologia, 21, 27, 65, 66, 104
Tear de Jacquard, 27
Tempo de computação usado pelos algoritmos, 198–199
Tentativa de divisão, 214
Teorema Binomial Estendido, 487
Teorema Chinês do Resto, 235–237, 258
Teorema da Curva de Jordan, 288
Teorema das Quatro Cores, 668–671, 676
Teorema de Bayes, 417–425
  generalização do, 420
Teorema de Dirac, 641
Teorema de Kleene, 819–821, 839
Teorema de Kuratowski, 663–665, 676
Teorema de Lamé, 298
Teorema de Ore, 641, 647
Teorema de Pick, 292
Teorema de Wilson, 244
Teorema do aperto de mãos, 599
Teorema dos números primos, 213, 584
Teorema Fundamental da Aritmética, 211, 258
Teorema Multinomial, 382
Teorema(s), 75, 105
  Aperto de Mãos, do, 599
  Binômio de Newton (Teorema binomial), 363–365, 387
  Binômio de Newton (Teorema binomial) estendido, 487
  Chinês do Resto, 235–237, 258
  Curva de Jordan, 288
  da Galeria de arte, 675
  das Quatro Cores, 668–671, 676
  de Bayes, 417–425
  de demonstração, automatizada, 108
  de Dirac, 641
  de Kleene, 819–821, 839
  de Kuratowski, 663–665, 676
  de Lamé, 298
  de Ore, 641, 647
  de Pick, 292
  de propriedade arquimediana, 849
  de Wilson, 244
  do Número Primo, 213
  Fundamental da Aritmética, 211, 258, 285–286
  Master, 479
  métodos de demonstrações, 76–83
  Multinomial, 382
  pequeno de Fermat, 239, 258
  último de Fermat, 100–101, 299
Teoria básica de conjuntos, 112
Teoria de probabilidade, 393, 400–416
Teoria de Ramsey, 352
Teoria dos números, 200–244
Terminais, 787
Termo inicial, 149, 150, 151
Terra, 424
Tese de Church–Turing, 833
Teste
  de Lucas–Lehmer, 212, 213
  de Miller, 245
  de probabilidade, 412–413
  de Turing, 27
  probabilístico para verificação se um número é primo, 412–413
Teste de Bernoulli, 428, 442
  distribuição binomial, 406–408
  mutuamente independente, 406
Teste de drogas, 424
Teste de Miller para base *b*, 245
Teste para verificação se um número é primo, 412–413
Teste probabilístico para verificação se um número é primo, 412–413
Testes mutuamente independentes, 406
Tetraominós, 104
TeX, 184
The Canterbury Puzzles (Dudeney), 454
Tipo de dado, 113
Topologia anel para redes locais, 605
Topologia estrela para redes locais, 605
Topologia híbrida para redes de área local, 605
Topologia para redes locais
  anel, 605
  estrela, 605
  híbrido, 605
Torneio, 680
Torneio Round-robin, 291, 416, 593, 594
Toro, 666
Torre de Hanói, 452–454
Trabalhos, distribuição de, 513,
Traça, 811
Tradução
  de expressões lógicas para sentenças do português, 42–43
  sentenças em português para expressões lógicas, 10–11, 42–43, 56–57
Transformação afim, 208
Transitividade universal, 74
Transmissão de vírus de computador, 389
Transmissão múltipla, 726
Transporte, 222
Transposição de matrizes, 251, 257
Traversa de árvore, 710–722, 743
  em ordem simétrica, 712, 714, 715, 718, 743
  ordem de nível, 745
  pós-ordem, 712, 715–716, 717, 718, 743–744
  pré-ordem, 712–713, 718, 743
Três pontos (. . .), 112
Triangulação, 288
Triângulo de Pascal, 366, 386
Trilha, 622
Triominó, 100, 277, 283
  à direita, 100, 277, 283
  reto, 100
Tripla de Hoare, 323
Truques de mágica, 14
Tukey, John Wilder, 16
Turing, Alan Mathison, 27, 176, 197, 825, 826
Ulam, Stanislaw, 483
Último Teorema de Fermat, 100–101, 299
União
  de conjuntos, 121, 126–128, 163
  de conjuntos finitos de dois, número de elementos em, 500, 514
  de conjuntos finitos de três, número de elementos em, 501–503, 514
  de conjuntos fuzzy, 133
  de grafos, 661, 676
  de multiconjuntos, 132
Unicórnios, 17
Unidade, fração egípcia, 331
UNIVAC, 811
Universo de discurso, 34, 105
Usando funções generalizadas para resolver problemas de contagem, 488–493
  relações de recorrência, 493–495
Valeé-Poussin, Charles-Jean-Gustave-Nicholas de la, 213

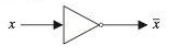
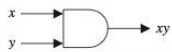
Validade
  argumento, 64
  forma de argumento, 64
Valor(es)
  de árvore, 706
  de vértice em árvores de jogo, 706–707, 743
  esperado(s), 426–429, 442
    de inversões em permutação, 430–431
    linearidade da esperança, 429–431, 442
    no problema de verificação de chapéus, 430
  final, 857
  inicial, 857
  threshold, 783
Valores esperados, 426–429, 442
  de inversões em permutação, 430–431
  linearidade de, 429–431, 442
  no problema do chapéu, 430
Valor-verdade, 104
  de implicação, 6
  de proposição, 3
Vandermonde, Alexandre-Théophile, 368
Variância, 426, 436–438, 442
Variáveis aleatórias, 402, 408–409
  covariância de, 441
  definição de, 442
  desvio-padrão de, 436
  distribuição de, 408, 442
    geométrica, 433–434
  independente, 434–436, 442
  indicador, 440
  valores esperados de, 402–403, 426–429, 442
  variância de, 426–429, 442
Variáveis aleatórias independentes, 434–436, 442
Variável(s)
  aleatória, 402, 408–409
  covariância de, 442
  definição de, 442
  desvio-padrão de, 436
  distribuição de, 408, 442
    geométrica, 433–434
  independente, 434–436, 442
  indicador, 440
  valores esperados de, 402–403, 426–429, 442
  variância de, 426–429, 442
  booleana, 15, 104, 750, 758, 781
  ligada, 38–39
  livre, 38, 105
Veitch, E. W., 768
Venn, John, 115
Verbo, 786
Verificação de programa, 323
Vértice(s), 541
  adjacente, 598, 600, 675
  ancestral de, 686, 743
  conexão, 598
  conjunto independente de, 680
  contando caminhos entre, 628–629
  de corte, 625
  de grafo não orientado, 592, 598, 601
    grau de, 598
  de grafo simples, 590, 612
  de grafos orientados, 591, 600
  de grau de, 600, 675
  de multigrafo, 590
  de multigrafo direto, 592
  de polígono, 288
  de pseudografo, 590
  descendente de, 686, 743
  distância entre, 680
  em grau de, 600, 675
  excentricidade de, 695
  filho de, 686, 687, 743
  final, 541, 600
  grau de, em grafo não orientado, 598
  inicial, 541, 600
  interior, 551
  interno, 686, 743
  irmão de, 686, 743
  isolado, 598, 676
  nível de, 691, 743
  número de, de árvore binária completa, 306
  ordem de nível de, 745
  pai de, 686, 743
  pendente, 598, 676
  valor de, em árvore de jogo, 706–707, 743
Vértice, 288
  bipartição de, 602
Vértice da aresta, 541
Vértice inicial, 541, 600
Vértice isolado, 598, 676
Vértice pendente, 598, 676
Vértice terminal, 541, 600
Vértices adjacentes, 598, 600, 675
Vértices conectados, 598
Vértices de corte, 625
Vértices interiores, 551
Vértices internos, 686, 744
Vizinhos em grafos, 598
Vocabulário, 787, 838
Votação majoritária, circuito para, 763
Warshall, Stephen, 551
WAVES, Navy, 811
Web, navegadores, 595
Web spiders, 734
Website para este livro, x, xvii
Whitehead, Alfred North, 114
Wiles, Andrew, 101, 239
Word and Object (Quine), 776
XML, 796
XOR, 15–16
Ziegler's Giant Bar, 184
Zodíaco, signos do, 388

Lista de Símbolos

TÓPICO	*SÍMBOLO*	*SIGNIFICADO*	*PÁGINA*
LÓGICA	$\neg p$	negação de p	3
	$p \wedge q$	conjunção de p e q	4
	$p \vee q$	disjunção de p e q	4
	$p \oplus q$	ou exclusivo entre p e q	5
	$p \rightarrow q$	a implicação p implica q	6
	$p \leftrightarrow q$	bicondicional de p e q	9
	$p \equiv q$	equivalência entre p e q	22
	$\mathbf{V}$	Tautologia	24
	$\mathbf{F}$	Contradição	24
	$P(x_1, \ldots, x_n)$	função proposicional	32
	$\forall x P(x)$	quantificação universal de $P(x)$	34
	$\exists x P(x)$	quantificação existencial de $P(x)$	36
	$\exists! x P(x)$	unicidade da quantificação de $P(x)$	37
	$\therefore$	portanto	63
	$p\{S\}q$	exatidão parcial de S	323
CONJUNTOS	$x \in S$	x é um elemento de S	112
	$x \notin S$	x não é um elemento de S	112
	$\{a_1, \ldots, a_n\}$	lista de elementos de um conjunto	112
	$\{x \mid P(x)\}$	lei de formação de um conjunto	112
	$\mathbf{N}$	conjunto dos números naturais	113
	$\mathbf{Z}$	conjunto dos números inteiros	113
	$\mathbf{Z}^+$	conjunto dos números inteiros positivos	113
	$\mathbf{Q}$	conjunto dos números racionais	113
	$\mathbf{R}$	conjunto dos números reais	113
	$S = T$	igualdade de conjuntos	113
	$\emptyset$	conjunto vazio (ou nulo)	114
	$S \subseteq T$	S é um subconjunto de T	114
	$S \subset T$	S é um subconjunto estrito de T	115
	$\lvert S \rvert$	cardinalidade de S	116
	$P(S)$	o conjunto das partes de S	116
	$(a_1, \ldots, a_n)$	n-upla	117
	(a, b)	par ordenado	117
	$A \times B$	produto cartesiano de A e B	118
	$A \cup B$	união de A com B	121
	$A \cap B$	interseção de A e B	121
	$A - B$	a diferença entre A e B	123
	$\overline{A}$	complemento de A	123
	$\bigcup_{i=1}^{n} A_i$	união de A_i, $i = 1, 2, \ldots, n$	127
	$\bigcap_{i=1}^{n} A_i$	interseção de A_i, $i = 1, 2, \ldots, n$	128
	$A \oplus B$	diferença simétrica entre A e B	131

TÓPICO	*SÍMBOLO*	*SIGNIFICADO*	*PÁGINA*
FUNÇÕES	$f(a)$	valor da função f em a	133
	$f: A \to B$	função de A em B	133
	$f_1 + f_2$	soma das funções f_1 e f_2	135
	$f_1 f_2$	produto das funções f_1 e f_2	135
	$f(S)$	imagem de S na função f	136
	$\iota_A(s)$	função identidade em A	138
	$f^{-1}(x)$	função inversa de f	139
	$f \circ g$	função composta de f e g	140
	$\lfloor x \rfloor$	maior inteiro menor que x	143
	$\lceil x \rceil$	menor inteiro maior que x	143
	a_n	termo de $\{a_i\}$ com índice n	150
	$\sum_{i=1}^{n} a_i$	soma de $a_1, a_2, \ldots, a_n$	153
	$\sum_{\alpha \in S} a_\alpha$	soma de a_α para todo $\alpha \in S$	156
	$\prod_{i=1}^{n} a_n$	produto de $a_1, a_2, \ldots, a_n$	162
	$f(x)$ é $O(g(x))$	$f(x)$ é O grande de $g(x)$	180
	$n!$	n fatorial	185
	$f(x)$ é $\Omega(g(x))$	$f(x)$ é Ômega grande de $g(x)$	189
	$f(x)$ é $\Theta(g(x))$	$f(x)$ é o Teta grande de $g(x)$	189
	$\sim$	comportamento assintótico	192
	$\min(x, y)$	o mínimo entre x e y	216
	$\max(x, y)$	o máximo entre x e y	217
	$\approx$	aproximadamente igual a	395
INTEIROS	$a \mid b$	a divide b	201
	$a \nmid b$	a não divide b	201
	a **div** b	o quociente inteiro da divisão de a por b	202
	a **mod** b	o resto inteiro da divisão de a por b	202
	$a \equiv b \pmod{m}$	a é congruente a b módulo m	203
	$a \not\equiv b \pmod{m}$	a não é congruente a b módulo m	203
	$\text{mdc}(a, b)$	máximo divisor comum entre a e b	215
	$\text{mmc}(a, b)$	mínimo múltiplo comum entre a e b	217
	$(a_k a_{k-1} \ldots a_1 a_0)_b$	representação na base b	219
MATRIZES	$[a_{ij}]$	matriz com entradas (elementos) a_{ij}	247
	$\mathbf{A} + \mathbf{B}$	matriz soma de **A** com **B**	247
	$\mathbf{AB}$	matriz produto de **A** com **B**	248
	$\mathbf{I}_n$	matriz identidade de ordem n	251
	$\mathbf{A}^t$	transposta de **A**	251
	$\mathbf{A} \vee \mathbf{B}$	matriz conjunção de **A** e **B**	252
	$\mathbf{A} \wedge \mathbf{B}$	matriz encontro de **A** e **B**	252
	$\mathbf{A} \odot \mathbf{B}$	produto booleano de **A** com **B**	253
	$\mathbf{A}^{[n]}$	n-ésima potência booleana de **A**	254

TÓPICO	*SÍMBOLO*	*SIGNIFICADO*	*PÁGINA*
CONTAGEM	$P(n, r)$	número de r-permutações de um conjunto com n elementos	355
	$C(n, r)$	número de r-combinações de um conjunto com n elementos	357
	$\binom{n}{r}$	coeficiente binomial n, r	357
	$C(n; n_1, n_2, \ldots, n_m)$	coeficiente multinomial	382
	$p(E)$	probabilidade de E	394
	$p(E \mid F)$	probabilidade condicional de E dado F	404
	$E(X)$	valor esperado da variável aleatória X	426
	$V(X)$	variância da variável aleatória X	436
	C_n	números de Catalan	456
	$N(P_{i_1} \ldots P_{i_n})$	número de elementos com as propriedades $P_{i_j}, j = 1, \ldots, n$	506
	$N(P'_{i_1} \ldots P'_{i_n})$	número de elementos sem as propriedades $P'_{i_j}, j = 1, \ldots, n$	506
	D_n	número de permutações caóticas de n objetos	510
RELAÇÕES	$S \circ R$	composição das relações R e S	526
	R^n	n-ésima potência da relação R	526
	R^{-1}	relação inversa	528
	s_C	operador de seleção para o condicional C	532
	$P_{i_1, i_2, \ldots, i_m}$	projeção	533
	$J_p(R, S)$	conjunção	534
	Δ	relação diagonal	545
	R^*	relação de conectividade de R	547
	$[a]_R$	classe de equivalência de a com relação a R	558
	$[a]_m$	classe de congruência módulo m	558
	(S, R)	estrutura formada por um conjunto S e uma relação de ordem parcial R	566
	$a \prec b$	a é menor que b	567
	$a \succ b$	a é maior que b	567
	$a \preccurlyeq b$	a é menor que ou igual a b	567
	$a \succcurlyeq b$	a é maior que ou igual a b	567
GRAFOS E ÁRVORES	(u, υ)	aresta orientada	541
	$G = (V, E)$	grafo com conjunto de vértices V e conjuntos de arestas E	589
	$\{u, \upsilon\}$	aresta não orientada	590
	$\mathrm{gr}(\upsilon)$	grau do vértice υ	598
	$\mathrm{gr}^-(\upsilon)$	grau de entrada do vértice υ	600
	$\mathrm{gr}^+(\upsilon)$	grau de saída do vértice υ	600
	K_n	grafo completo de n vértices	601
	C_n	ciclo de tamanho n	601
	W_n	roda de tamanho n	601
	Q_n	n-cubo	602
	$K_{m, n}$	grafo bipartido completo de tamanho m, n	604

TÓPICO	*SÍMBOLO*	*SIGNIFICADO*	*PÁGINA*
GRAFOS E ÁRVORES (cont.)	$G_1 \cup G_2$	união de G_1 e G_2	608
	$a, x_1, \ldots, x_{n-1}, b$	caminho de a para b	623
	$a, x_1, \ldots, x_{n-1}, a$	circuito	623
	r	número de regiões de um plano	660
	gr (R)	grau da região R	661
	n	número de vértices de uma árvore com raiz	690
	i	número de vértices internos de uma árvore com raiz	691
	l	número de folhas de uma árvore com raiz	691
	m	maior número de filhos de um vértice interno de uma árvore com raiz	686
	h	altura de uma árvore com raiz	692
ÁLGEBRA BOOLEANA	$\bar{x}$	complemento de uma variável booleana x	749
	$x + y$	soma booleana de x com y	749
	$x \cdot y$ (ou xy)	produto booleano de x com y	749
	B	$\{0, 1\}$	750
	F^d	dual de F	754
	$x \mid y$	x NE y	759
	$x \downarrow y$	x NOU y	759
		inversor	761
		porta *OU*	761
		porta *E*	761
LINGUAGENS E MÁQUINAS DE ESTADO FINITO	λ	cadeia vazia	151
	xy	concatenação de x e y	300
	$l(x)$	comprimento da cadeia x	301
	w^R	reverso de w	309
	(V, T, S, P)	gramática de estrutura de frase	787
	S	símbolo inicial	787
	$w \rightarrow w_1$	produção	787
	$w_1 \Rightarrow w_2$	w_2 é diretamente derivável de w_1	787
	$w_1 \overset{*}{\Rightarrow} w_2$	w_2 é derivável de w_1	787
	$< A > :: = < B > c \mid d$	forma de Backus–Naur	792
	(S, I, O, f, g, s_0)	máquina de estado finito com saída	798
	s_0	estado inicial	798
	AB	concatenação dos conjuntos A e B	804
	A^*	fecho de Kleene de A	804
	(S, I, f, s_0, F)	máquina de estado finito sem saída	805
	(S, I, f, s_0)	máquina de Turing	828